On grouping theory in dot patterns, with applications to perception theory and 3D inverse geometry

José Lezama

École Normale Supérieure de Cachan, France
Universidad de la República, Uruguay

On grouping theory in dot patterns, with applications to perception theory and 3D inverse geometry

A dissertation presented
by

José Lezama

in fulfillment of the requirements
for the degree of Doctor of Philosophy
in the subject of Applied Mathematics

Committee in charge

Referees	Yann GOUSSEAU	-	Télécom ParisTech, FR
	Guillermo SAPIRO	-	Duke University, USA
	Steven W. ZUCKER	-	Yale University, USA
Advisors	Rafael GROMPONE VON GIOI	-	ENS de Cachan, FR
	Jean-Michel MOREL	-	ENS de Cachan, FR
	Gregory RANDALL	-	Universidad de la República, UY
Examiners	Julie DELON	-	Université Paris Descartes, FR
	Pablo MUSÉ	-	Universidad de la República, UY

March 2015

Version of March 18, 2015 at 14:01.

Abstract of the Dissertation

José Lezama

On grouping theory in dot patterns, with applications to perception theory and 3D inverse geometry

Under the direction of:
Rafael Grompone von Gioi, Jean-Michel Morel and Gregory Randall

This thesis studies two mathematical models for an elementary visual task: the perceptual grouping of dot patterns. The first model handles the detection of perceptually relevant arrangements of collinear dots. The second model extends this framework to the more general case of good continuation of dots. In both cases, the proposed models are scale invariant and unsupervised. They are designed to be robust to noise, up to the point where the structures to detect become mathematically indistinguishable from noise. The experiments presented show a good match of our detection theory with the unmasking processes observed by gestaltists in human perception, supporting their perceptual plausibility.

The proposed models are based on the *a contrario* framework, a formalization of the non-accidentalness principle in perception theory. This thesis makes two contributions to the *a contrario* methodology. One is the introduction of adaptive detection thresholds that are conditional to the structure's local surroundings. The second is a new refined strategy for resolving the redundancy of multiple meaningful detections.

Finally, the usefulness of the collinear point detector as a general pattern analysis tool is demonstrated by its application to a classic problem in computer vision: the detection of vanishing points. The proposed dot alignment detector, used in conjunction with standard tools, produces improved results over the state-of-the-art methods in the literature.

Aiming at reproducible research, all methods are submitted to the IPOL journal, including detailed descriptions of the algorithms, commented reference source codes, and online demonstrations for each one.

Résumé de la Thèse

José Lezama

Sur la théorie du regroupement de points en 2D avec applications
à la théorie de la perception et à la géométrie 3D inverse:
Sous la direction de:
Rafael Grompone von Gioi, Jean-Michel Morel et Gregory Randall

Cette thèse porte sur l'étude de deux modèles mathématiques pour une tâche visuelle élémentaire: le regroupement perceptuel de points 2D. Le premier modèle traite la détection d'alignements de points. Le deuxième modèle étend ce cadre au cas plus général de la bonne continuation de points. Dans les deux cas, les modèles proposés sont invariants au changement d'échelle, et non supervisés. Ils sont conçus pour être robustes au bruit, jusqu'au point où les structures à détecter deviennent mathématiquement impossibles à distinguer du bruit. Les expériences présentées montrent une cohérence entre notre théorie de détection et les processus de démasquage observés par les gestaltistes dans la perception humaine.

Les modèles proposés sont basés sur la méthodologie *a contrario*, une formalisation du principe de non fortuité (*non-accidentalness principle*) dans la théorie de la perception. Cette thèse fait deux contributions au méthodes *a contrario*. L'une est l'introduction de seuils de détection adaptatifs qui sont conditionnels au voisinage des structures évaluées. La deuxième contribution est une nouvelle stratégie raffinée pour résoudre la redondance de plusieurs détections significatives.

Finalement, l'utilité du détecteur d'alignements de points comme outil général d'analyse de données est démontrée avec son application a un problème classique en vision par ordinateur: la détection de points de fuite. Le détecteur d'alignements de points proposé, utilisé avec des outils standards, produit des résultats améliorant l'état de l'art.

Visant à la recherche reproductible, toutes les méthodes sont soumises au journal IPOL, en incluant descriptions détaillées des algorithmes, du code source commenté et démonstrations en ligne pour chaque méthode.

Resumen de la Tesis

José Lezama

Sobre teoría de agrupamiento en patrones de puntos, con aplicaciones en teoría de la percepción y en geometría 3D inversa

Bajo la dirección de:
Rafael Grompone von Gioi, Jean-Michel Morel y Gregory Randall

Esta tesis estudia dos modelos matemáticos para una tarea visual elemental: el agrupamiento perceptual de patrones de puntos. El primer modelo trata la detección de arreglos colineales de puntos perceptualmente relevantes. El segundo modelo extiende este marco al caso más general de buena continuación de puntos. En ambos casos, los modelos propuestos son invariantes a cambios de escala y no supervisados. Estos modelos son diseñados para ser robustos al ruido, hasta el punto en que las estructuras para detectadar se vuelvan matemáticamente indistinguibles del ruido. Los experimentos presentados muestran una buena coherencia entre nuestra teoría de detección y los procesos de desenmascaramiento observados por gestaltistas en la percepción humana.

Los modelos propuestos son basados en la metodología *a contrario*, una formalización del principio de no-accidentalidad en teoría de la percepción. Esta tesis hace dos contribuciones a esta metodología. Una es la introducción de umbrales de detección adaptativos que son condicionales al entorno de las estructuras evaluadas. La segunda contribución es una nueva y refinada estrategia para resolver la redundancia de múltiples detecciones significativas.

Finalmente, la utilidad del detector de puntos alineados como herramienta general de análisis de patrones es demostrada con su aplicación a un problema clásico en vision por ordenador: la detección de puntos de fuga. El detector de puntos alineados propuesto, utilizado en conjunto con herramientas estándar, produce resultados mejores que el estado del arte en la literatura.

Con el objetivo de hacer investigación reproducible, todos los métodos son submitidos al journal IPOL, incluyendo descripciones detalladas de los algoritmos, código fuente comentado y demostraciones en línea para cada uno de ellos.

to Natalie and Pablo

Acknowledgments

My most sincere gratitude to my three advisors. To Jean-Michel Morel, whose generosity and wisdom are the pillars of this thesis. To Gregory Randall I will always be grateful for giving me the opportunity of embarking in this adventure. To my friend Rafael Grompone von Gioi, this ship would have not reached good port were it not for its never ending enthusiasm. Thanks for keeping the beacon alight.
A very special thanks to Yann Gousseau, Guillermo Sapiro and Steven W. Zucker for accepting to review my thesis and providing me with extremely encouraging feedback. Also, to Julie Delon and Pablo Musé for giving me the honor of accepting to be part of the jury.
To all the staff both in CMLA and IIE, Véronique Almodovar, Micheline Brunetti, Virginie Pauchont at CMLA and Maria Misa at IIE.
To all my friends and colleagues who have accompanied me during this years; Ariane, Barbara, Carlo, Cecilia, Enric, Federico, Gabriele, Germán, Gordo, Haldo, Irène, Ives, Javier, Juliana, Lara, Mariano, Martín, Matías, Mauricio, Michael, Miguel, Morgan, Nacho, Nicola, Nicolas, Norberto, Rachel, Sambita, Samy, Yi Qing, Zhong Wei and many others.
To my parents, my brother and my sister, and to all my supporting family. A special consideration to my French relatives, who have helped me enormously in my adopting country; Luis, Claudia, Gaïa, Pablo and Elvire. Also to my late grandparents, for opening the windows of French culture to me.
To my friends in Uruguay, I regret not being able to see you more often during these years.
Finally, to my dear wife Natalie, thank you for your unconditional support and for our beloved son Pablo.

Contents

1 Introduction

Motivation

Alan Turing advanced a controversial proposal in 1950 that is now known as the *Turing Test* [Turing 1950]. Turing's aim was to discuss the problem of machine intelligence and, instead of giving a premature definition of thinking, he framed the problem in what he called the *Imitation Game*: A human interrogator interacts with another human and a machine, but only in typewritten form; the task of the interrogator is to ask questions in order to determine which of its two interlocutors is the human. Turing proposed that a machine that eventually could not be distinguished from humans by its answers should be considered intelligent. This influential suggestion sparked a fruitful debate that continues to this day [Pinar Saygin et al. 2000].

Our concern here is however slightly different. We are studying perception and Turing precluded in his test any machine interaction with the environment other that the communication through the teletype; he concentrated on the pure problem of thinking and to that aim avoided fancy computer interactions, that anyway did not exist at his time. Yet, machine perception is still a hard problem for which current solutions are far from the capacities of humans or animals. Our purpose is to discuss a variety of *gestaltic perceptual imitation games* as a research methodology to develop machine vision algorithms on the one hand, and to open the way to quantitative psychophysical protocols on the other.

The Gestalt school, Wertheimer [1923]; Köhler [1947]; Ellis [1967 (originally 1938)]; Metzger [1975]; Kanizsa [1980] among others, developed from the twenties to the eighties an original *modus operandi*, based on the invention and display to subjects of clever geometric figures [Wagemans et al. 2012a,b]. A considerable mass of experimental evidence was gathered, leading to the conclusion that the first steps of visual perception are based on a reduced set of geometrical grouping laws. Unfortunately these Gestalt laws, relevant though they were, remained mainly qualitative and led to no direct machine perception approach.

Since the emergence of the field of Computer Vision [Marr 1982] about fifty years ago – initially as a branch of the Artificial Intelligence working with robots and its artificial senses – there have been many attempts at formalizing vision theories and especially Gestalt theory [Sarkar and Boyer 1993]. Nevertheless, only a small fraction of these proposals has been accompanied by systematic efforts to compare machine and human vision. An important exception is the Bayesian theory of perception [Mumford 2002] that has attracted considerable attention in cognitive sciences, leading to several experimental evaluations [Feldman 2001; Kersten et al. 2004].

Such experiments stress the relevance of computer vision as a research program in vision, in addition to a purely technological pursuit. Its role should be complementary to explanatory sciences of natural vision by providing, not only descriptive laws, but actual implementations of

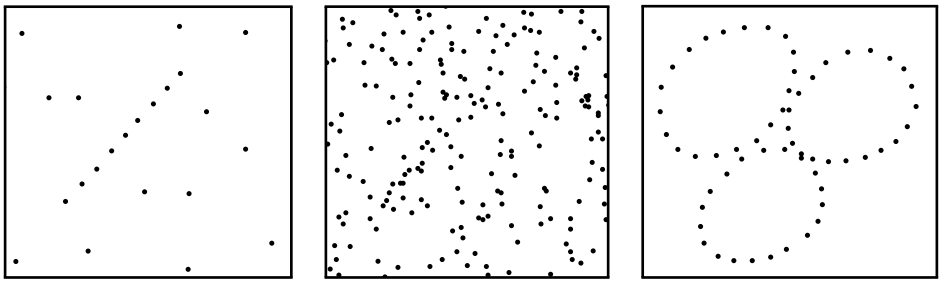


Figure 1.1: Illustration of the complexity of the point alignment problem: Exactly the same set of aligned dots is present in the three images, but it is only perceived as such in the first one. The second one is a classic "masking by texture" case and the third shows a "masking by structure", often called "Gestalt conflict".

mechanisms of operation. With that aim, perceptual versions of the imitation game should be the *Leitmotiv* in the field, guiding the conception, evaluation and success of theories.

One of the procedures used by Gestalt psychology practitioners was to create clever geometric figures that would reveal a particular aspect of perception when used in controlled experiments with human subjects. They pointed out the grouping mechanisms, but also the striking fact that geometric structures objectively present in the figure are not necessarily part of the final gestalt interpretation. These figures are in fact counterexamples against simplistic perception mechanisms. Each one represents a challenge to a theory of vision that should be able to cope with all of them.

The methodology we followed in order to design and improve automatic geometric gestalt detectors is in a way similar to that of the gestaltist. One starts with a primitive method that works correctly in very simple examples. The task is then to produce data sets where *humans* clearly see a particular gestalt while the rudimentary method produces a different interpretation. Analyzing the errors of the first method gives hints to improve the procedure in order to create a second one that produces better results with the whole data set produced until that point. The same procedure is applied to the second method to produce a third one, and successive iterations refine the methods step by step. The methodology used by the Gestalt psychologist to study human perception is used here to push algorithms to be similar to their natural counterpart. Finding counterexamples is less and less trivial after some iterations and the counter-examples become, like gestaltic figures, more and more clever. Each detection game will only stop when it eventually passes the Turing test, the algorithm's detection capability becoming indistinguishable from that of a human.

In this thesis we will attempt at modeling simple tasks of human perception with algorithms based on the *non-accidentalness principle* introduced by Witkin, Tenenbaum and Lowe [Lowe 1985; Witkin and Tenenbaum 1983a,b] as a general grouping law. This principle states that spatial relations are perceptually relevant only when their accidental occurrence is unlikely. We shall use the *a contrario* framework, a particular formalization of the principle due to Desolneux, Moisan and Morel [Desolneux et al. 2003, 2008] as part of an attempt to provide a mathematical foundation to Gestalt Theory.

We will start with the study of a very basic visual task: the grouping of collinear dots in dot patterns. We will build increasingly complex versions of the algorithm that can handle masking situations, up to the point where other gestalts are required to explain a grouping. Next, we will attempt at modeling a more general case, the grouping of dots by good continuation. Through qualitative and quantitative experimentation, we will show the perceptual plausibility of our methods, and successfully compare it to other methods in the literature. Finally, to demonstrate the applicability

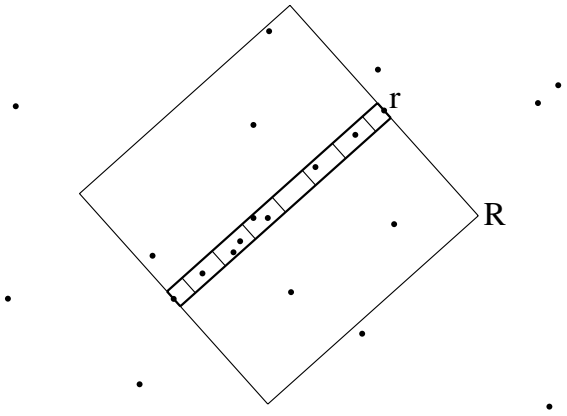


Figure 1.2: Representation of the candidate rectangle of the point alignment detector. In this case, the candidate rectangle r is divided into $c = 6$ boxes, 5 of which are occupied. The local density estimation window is R.

of these pattern analysis algorithms to practical problems, we use the dot alignment detector for the problem of vanishing point detection, obtaining results that improve the state-of-the-art.

1.1 Point Alignments Detection

From a gestaltic point of view, a point alignment is a group of points sharing the property of being aligned in one direction. While it may seem a simple gestalt, Figure 1.1 shows how complex the alignment event is. From a purely factual point of view, the same alignment of points is present in the three figures. It is however only perceived as such in the first figure by most viewers. The second the third figures illustrate two types of masking: the *masking by texture*, which occurs when a gestalt is surrounded by a clutter of randomly distributed *distractors*, and the *masking by structure*, which happens when the alignment is masked by other perceptually more relevant gestalts, a phenomenon also called *perceptual conflict* by gestaltists [Metzger 1975, 2006 (originally 1936); Kanizsa 1980]. The magic disappearance of the alignment in the second and third figures can be accounted for in two very different ways. As for the first one, we shall see that a probabilistic *a contrario* model [Desolneux et al. 2008] is relevant and can lead to a quantitative prediction. As for the second disappearance, it requires the intervention of another more powerful grouping law, the *good continuation* [Kanizsa 1979], which will be addressed in Chapter 3.

These examples show that a mathematical definition of dot alignments is required before even starting to discuss how to detect them. A purely geometric-physical description is clearly not sufficient. Indeed, an objective observer making use of a ruler would be able to state the existence of the very same alignment on all three figures. But this statement contradicts our perception, as it would contradict any reasonable computational perceptual theory.

This experiment also shows that the detection of an alignment is highly dependent on the context of the alignment. It is therefore a complex question, and must be decided by building mathematical definitions and detection algorithms, and confronting them to perception. As the patterns of Figure 1.1 already suggest, simple computational definitions with increasing complexity will nevertheless find perceptual counterexamples. In Chapter 2 of this thesis we present a series of algorithms with incrementing sophistication for detecting alignments of points in a point pattern.

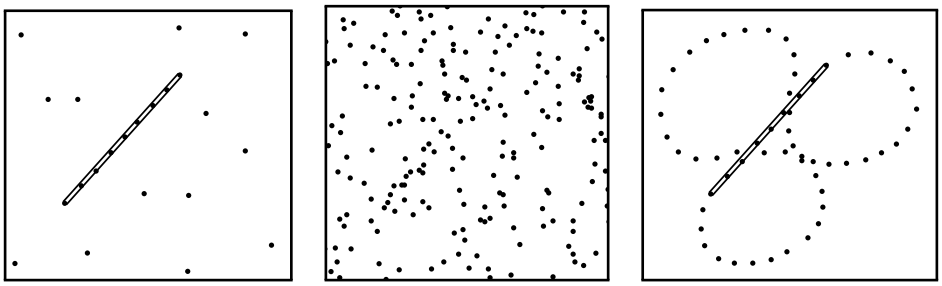


Figure 1.3: Results of the alignment detector in the examples of Figure 1.1. The first and second examples are correctly solved. The third, however requires the intervention of more powerful grouping laws, namely the good continuation and closure laws.

The two key aspects of the alignment will be shown to be its local density and its regularity. Our final method combines both criteria into a single coherent detector.

Our model is based on using a template consisting of a thin rectangle inside a larger one (see Figure 1.2). The inner rectangle is divided into equal boxes and the number of boxes occupied by at least one point is counted. The number of occupied boxes is then compared to its expectation under a null hypothesis of random and independently distributed points. This constitutes the Number of False Alarms (NFA), the fundamental quantity in the *a contrario* theory. The NFA in this case is expressed as

$$\mathrm{NFA}(r, R, c, \mathbf{x}) = N_{tests} \cdot \mathbb{P}\Big[b(r, c, \mathbf{X}) \geq b(r, c, \mathbf{x}) \,\Big|\, n(R, \mathbf{X}) = n(R, \mathbf{x})\Big]. \tag{1.1}$$

Here, $\mathbb{P}$ denotes a probability, b is the number of occupied boxes out of a total of c boxes, r and R are the inner and outer rectangles respectively, and $\mathbf{x}$ and $\mathbf{X}$ are the observed points and a random set of N points following the null hypothesis, respectively. The number n counts the points inside the outer rectangle. N_{tests} is the total number of tests computed (i.e. the number of templates tried). It will be obtained by considering rectangles formed by every pair of points and a given set of widths for the inner and outer rectangle, and a given set of number of boxes. In Chapter 2 we will prove that the NFA is a bound on the expectation of occurrence of the event under the null hypothesis. We will set a detectability threshold for the NFA of 1, meaning that under the background model, less than one false alarm would be obtained in average. We propose two contributions to the *a contrario* framework. One is the use of sophisticated conditional events on random point sets, for which expectation we nevertheless find easy bounds. Second is a new formulation of the exclusion principle to avoid redundant detections, based on comparing meaningful detections in a one to one basis.

Results of the final alignment detector are shown in Figures 1.3 and 1.4. In Figure 1.3, the detector correctly solves the first two figures, finding the trivial alignment in the first one and determining that there is visual masking in the second one. In the third figure however, the result of the algorithm is not the most natural explanation on the figure. Here there are clearly other gestalts involved, which produce a more relevant percept, for example the Gestalt laws of good continuation and closure. The law of good continuation will be treated in Chapter 3. In Figure 1.4 the surrounding rectangle and small boxes are also shown, as well as the logarithm of the NFA for each detection (best viewed in electronic format). In Chapter 4 we will use this alignment detector as a general pattern analysis tool to find vanishing points in urban scenes.

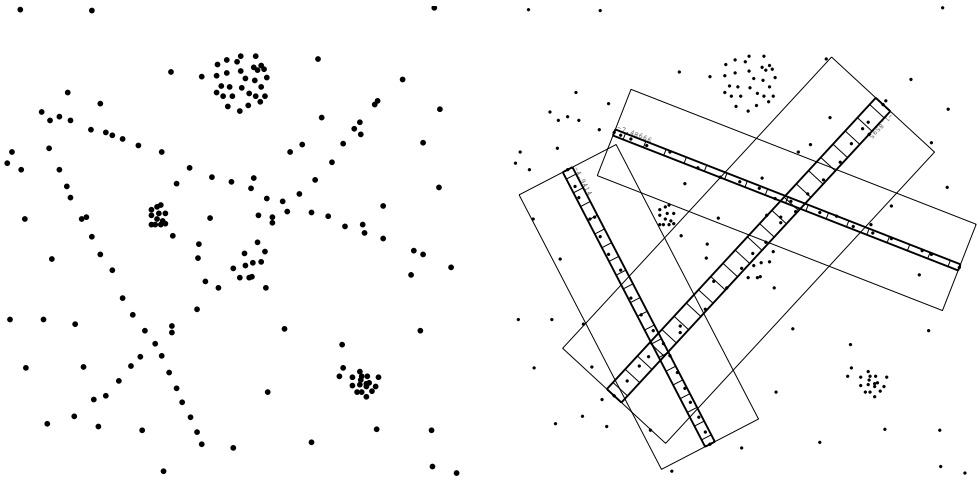


Figure 1.4: Result of the point alignment detector. Note how it correctly finds perceivable alignments while not being fooled by clusters and background noise.

This work has led to the publications of the following articles: Lezama et al. [2014c], Lezama et al. [2014d] and Lezama et al. [2014e].

1.2 Good Continuation Detection

A more general visual task is the grouping of dots by good continuation. The good continuation law was one of the first to be enunciated by gestaltists [Wertheimer 1923] and can be stated as "All else being equal, elements that can be seen as smooth continuations of each other tend to be grouped together" [Palmer 1999]. This phenomena is exemplified in Figure 1.5, where (despite the familiar shape of the letters) human perception easily finds the structures forming smooth and regularly spaced curves of dots. Besides receiving great interest in psychophysics [Uttal 1973; Prinzmetal and Banks 1977; Caelli et al. 1978; van Oeffelen and Vos 1983; Smits et al. 1984; Feldman 1997b; Pizlo et al. 1997], there have also been several proposals for formalizing this law since the early days of computer vision [Grossberg and Mingolla 1987; Sha'asua and Ullman 1988; Parent and Zucker 1989; Gigus and Malik 1991; Guy and Medioni 1992; Williams and Thornber 1999].

In Chapter 3 we will propose a model for the grouping of dots by good continuation, inspired by the findings of the grouping of aligned dots. Our good continuation model is based on local symmetries and, as before, the non-accidentalness principle to determine perceptually relevant configurations of good continuation of dots. We will also derive a grouping algorithm based on this model, an example result of which can be seen in Figure 1.5 (right).

Given three points, our model will consider the distance of the third point to the symmetric of the first with respect to the second (see Figure 1.6(b)). In this case, the probabilistic event is defined as the probability of the event defined as: "Given n points observed inside the big circle of radius R, the closest point to the ideal symmetric X is closer than r". This is expressed as

$$p = \mathbb{P}(r_{NN} \leq r) = 1 - \left(1 - \frac{r^2}{R^2}\right)^n. \tag{1.2}$$

Next, we will consider chains formed by triplets of points, where each subsequent triplet shares

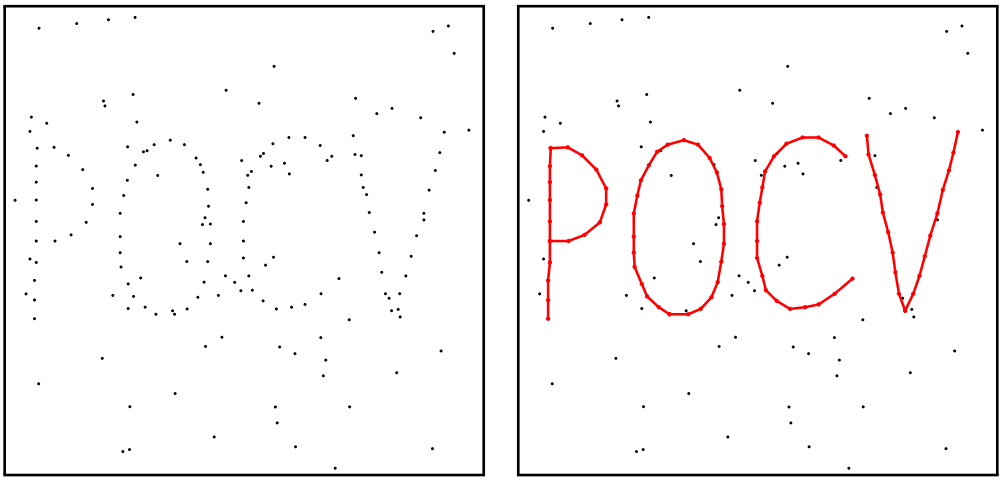


Figure 1.5: Illustration of perceptual grouping by the good continuation law. A human observer tends to group elements among smooth and long curves. Experimental results for the *good continuation* detector. Left: original points. Right: most meaningful good continuation groupings obtained by our algorithm. Note that as a general clustering method, the algorithm automatically determines the number of structures, and is able to distinguish structure from noise.

two points with the previous one. Assuming independence of the points location, the probability of occurrence of a chain under the random model will be computed as the multiplication of each triplet probability. Finally, the NFA of a good continuation event of K points will be stated as

$$\text{NFA} = N_{tests} \cdot p_{\max}^{K-2}. \tag{1.3}$$

Here, $p_{\max}$ represents the worst observed precision, so the event is expressed as "observing a chain of points where all triplets have a precision at least as good as $p_{\max}$". The number N_{tests} is again the number of all chains evaluated, and it will be obtained by considering triplets formed by points and their nearest neighbors. Again, in Chapter 3 we will prove that this NFA effectively bounds the expected number of occurrences of such an event under the background model of random, independent and uniformly distributed points. As in Chapter 2, we will set the detectability threshold of the NFA at 1, which means that on average less than one false detection would occur by chance.

We will also propose an efficient algorithm for detecting good continuation configurations in dot patterns. Candidate curves will be obtained by computing possible triplets using nearest neighbors, then building a graph representation of the triplets, where adjacencies are given by the sharing of two points, and edge costs by the sum of the triplets precisions. On this graph we will use a shortest path method (the Floyd-Warshall algorithm) as an heuristic to obtain candidates chains of triplets. To each candidate chain we will compute the NFA and we will keep only the most meaningful ones, applying a masking principle similar to the one introduced in Chapter 2.

We will demonstrate the perceptual relevance of our model with experiments on datasets taken from classical psychophysics articles, such as the work of Uttal [Uttal 1973]. Finally, we will compare the results of our algorithm in a structure/noise segmentation task with a well-established method in computer vision, the Tensor Voting approach [Guy and Medioni 1992, 1993; Mordohai and Medioni 2006]. The comparison yields that our method addresses three main drawbacks of Tensor Voting: scale invariance, robustness to noise and the lack of a principled criterion to extract final curves.

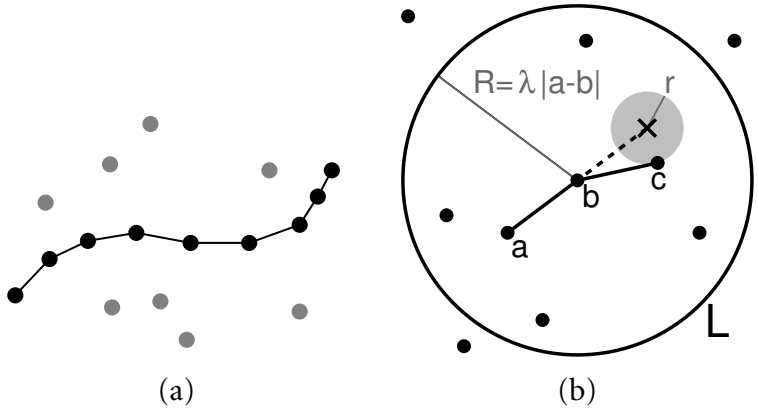


Figure 1.6: Definition of the *good continuation* event. **(a)** A candidate *chain* is defined by an ordered sequence of points. **(b)** Three consecutive points in a chain define a *triplet*, (a, b, c) in this case. Ideally, the triplet should be symmetric. The symmetry precision of a triplet is measured by the distance r from the ideal point X to its nearest point. When evaluated relative to a local window L of radius R, this precision can also be expressed as the probability that, among the n points in the local window, the nearest point to X be at most at a distance r.

Figure 1.7 shows the result of the good continuation detector for the three examples of Figure 1.1. The result for the first and second examples is perceptually correct. The result for the third example is better than the one of the point alignment detector, in the sense that it determines that the whole structure is perceptually relevant. The explanation of the figure is however not the most plausible one, as a human observer tends to see the three closed curves. To obtain that result we would need to incorporate other gestalts such as closure and convexity. It would also require a comparison of the meaningfulness as a global interpretation, between the combination of the three curves and a single long curve going through all points. This kind of comparison is not addressed in this thesis but it is clearly a necessary line of future research in computational Gestalt theory.

This work has led to the publications of the following articles: Lezama et al. [2014b] and Lezama et al. [2015a].

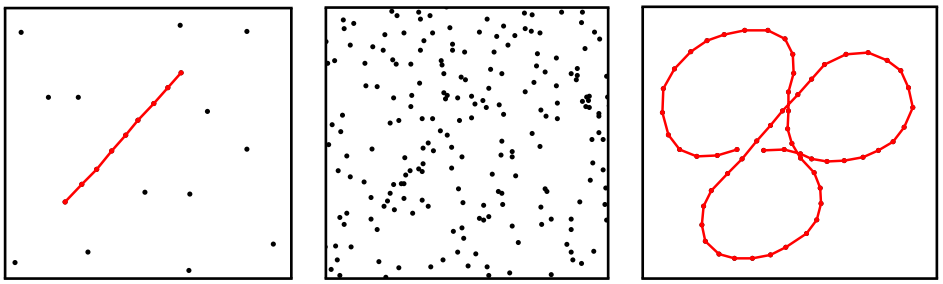


Figure 1.7: Results of the good continuation detector in the examples of Figure 1.1. The first and second examples are again correctly solved. In the third example, the result is successful in terms of finding the best curve. However, most viewers describe this figure as a set of three convex closed curves. In this case a comparison in meaningfulness of the combined three curves versus the long curve would be required.

1.3 Vanishing Points Detection

In the third part of this thesis, we will demonstrate the utility of the framework on practical problems. In particular, we will use the point alignment detector as a general, unsupervised and robust clustering technique for the classical computer vision problem of vanishing point detection. Our results improve the state-of-the-art [Xu et al. 2013] on two public and widely used datasets [Denis et al. 2008; Barinova et al. 2010].

The vanishing point detection problem can be briefly described as follows. Under the pinhole camera model, 3D lines in the space are transformed into 2D lines in the image. Moreover, parallel lines in 3D are projected into lines that converge on a point (perhaps at infinity) known as a *vanishing point* (see Figure 1.8). In the presence of parallel lines, as is common in human-made environments, vanishing points provide crucial information about the 3D structure of the scene

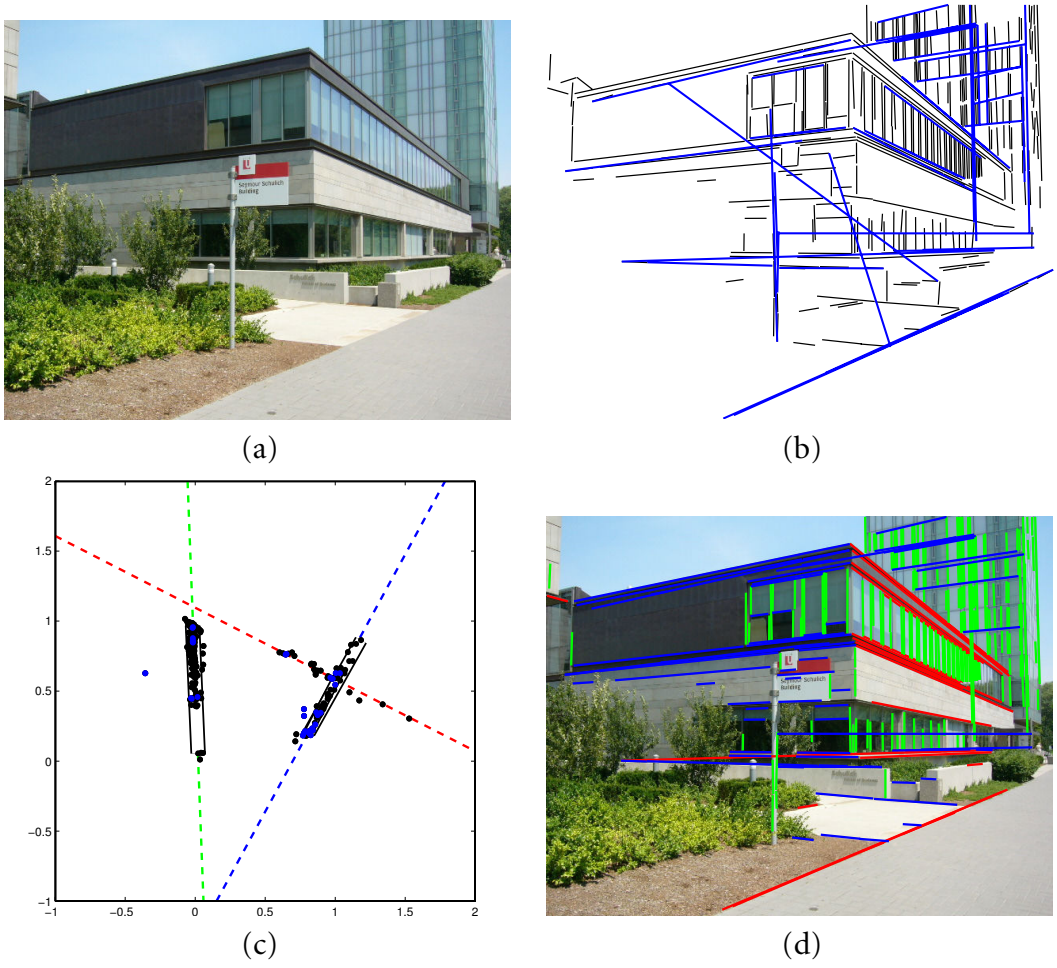


Figure 1.8: Brief representation of the steps involved in our vanishing point detection method. **(a)** Original image. **(b)** Detected line segments (black) and alignments of line segment endpoints (blue). **(c)** Mapping of the line segments using the PClines transform [Dubská et al. 2011], where converging lines become aligned points. Detected alignments are represented with two black lines. Colored dashed lines represent the ground truth vanishing points. **(d)** Resulting vanishing directions. Each line segment converging to one of the vanishing points is colored accordingly.

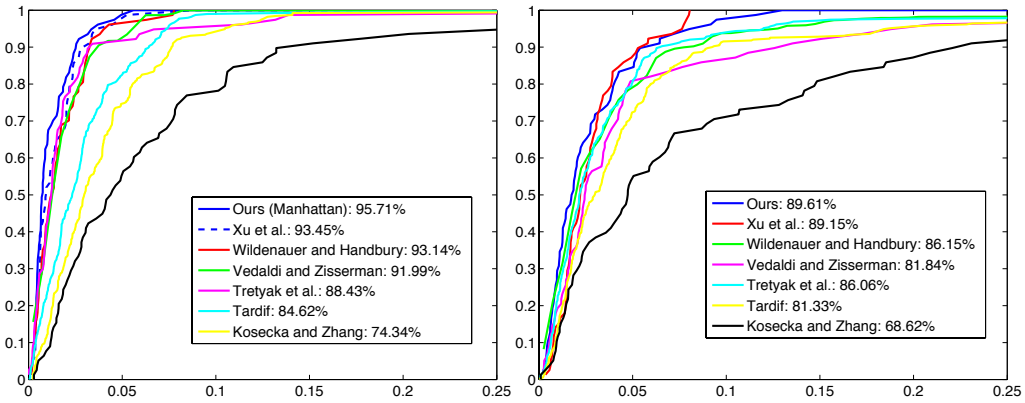
and have applications in camera calibration, single-view 3D scene reconstruction, autonomous navigation, and semantic scene parsing, to mention a few.

We will pose the vanishing point detection problem as a problem of finding clusters of aligned points. In order to do this, we use the recently introduced PClines transformation [Dubská et al. 2011], a line-to-point mapping that maps converging lines into aligned points. In fact, this is a perfect setting to apply the point alignment detector. Line segments in the image that do not correspond to a group of converging lines will become noise points surrounding the clusters. On the other hand, an image can have an arbitrary number of vanishing points. The unsupervised nature of the point alignment detection algorithm allows it to detect the number of clusters automatically.

We will also see that a second application of the alignment detector is possible in this problem. By finding alignments of line segment endpoints, we will introduce new cues for the vanishing directions. Imagine for example an image where only the vertical borders of a row of windows have been detected. By joining the endpoints of the line segments, new horizontal cues are obtained.

Figure 1.8 shows an example of the vanishing point detection procedure. In Figure 1.8(b), we show in blue the alignments of line segment endpoints found by the point alignment detector. In fact, thanks to the line segment endpoints alignments the method finds horizontal lines in the building in the back, which were missed by the line segment detector. In Figure 1.8(c), we show the dual space with the points resulting of the transformation of the line segments using the PClines transform. The colored dashed lines in the dual space represent points in the image corresponding to the ground truth vanishing points. As it can be seen, the converging segments in the image form elongated clusters in the dual space. These elongated clusters are correctly detected by our algorithm (the detections are represented by two black bars). Figure 1.8(d) shows in a different color the line segments corresponding to each vanishing point.

Once the vertical and horizontal vanishing points have been determined, one can estimate the position of the horizon line. In the case where there are multiple horizontal vanishing points, this estimation is done with a weighted average. In our method, we use the NFA of each detection as the weights. The error in the estimation of the horizon line is a quantitative performance measure widely used in the literature [Barinova et al. 2010; Wildenauer and Hanbury 2012; Vedaldi and Zisserman 2012; Xu et al. 2013]. Using this error measure, our method outperforms the state-



(a) York Urban Dataset (b) Eurasian Cities Dataset

Figure 1.9: Cumulative histograms of the horizon detection error for our method and competing methods. The horizontal axis represents the horizon line error. The vertical axis represents the ratio of images with horizon line error lower than the corresponding abscissa.

of-the-art [Xu et al. 2013] in two standard datasets of a hundred images each, the York Urban Dataset [Denis et al. 2008] and the Eurasian Cities Dataset [Barinova et al. 2010]. Figure 1.9 shows precision-recall curves for our method and competitors for both datasets.

This work has led to the publications of the following articles: Lezama et al. [2014a] and Lezama et al. [2015b].

List of Publications

The work in this thesis has led to the following publications:

- Lezama, J., Blusseau, S., Morel, J.M., Randall, G., and Grompone von Gioi, R. *Psychophysics, Gestalts and games*. In Sarti, A. and Citti, G., editors, Neuromathematics of Vision. Springer, 2014.
- Lezama, J., Morel, J.-M., Randall, G., and Grompone von Gioi, R. *A Contrario 2D Point Alignment Detection*. IEEE Transaction on Pattern Analysis and Machine Intelligence (TPAMI). Published online on August 5, 2014.
- Lezama, J., Randall, G., Morel, J.M., and Grompone von Gioi, R. *An Unsupervised Point Alignment Detection Algorithm*. Image Processing On Line (IPOL), submitted.
- Lezama, J., Grompone von Gioi, R., Randall, G., and Morel, J.M. *A contrario detection of good continuation of points*. In IEEE International Conference on Image Processing (ICIP) 2014.
- Lezama, J., Randall, G., Morel, J., and Grompone von Gioi, R. *Good continuation in dot patterns: a quantitative approach based on local symmetry and non-accidentalness*. Vision Research, submitted.
- Lezama, J., Grompone von Gioi, R., Randall, G., and Morel, J. M. *Finding vanishing points via point alignments in image primal and dual domains*. In Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 509–515, 2014.
- Lezama, J., Randall, G., Morel, J.-M., and Grompone von Gioi, R. *An Algorithm for Detecting Vanishing Points via Point Alignments*. Image Processing On Line, submitted.

List of Online Demos

The algorithms developed in this thesis can be tested in the following online demos:

- Point alignment detection:
  `http://dev.ipol.im/~jirafa/ipol_demo/point_alignment_detection/`
  (user: demo, password: demo)
- Good continuation detection:
  `http://dev.ipol.im/~jlezama/ipol_demo/lgrm_good_continuation_matlab/`
  (user: demo, password: demo)
- Vanishing points detection:
  `http://dev.ipol.im/~jlezama/ipol_demo/vanishing_points/`
  (user: demo, password: demo)

2 On Point Alignments Detection

In spite of many interesting attempts, the problem of automatically finding alignments in a 2D set of points seems to be still open. In this chapter, we shall first illustrate the difficulty of the problem by elementary examples and then propose an elaborate solution. We show that a correct alignment detection depends on not less than four interlaced criteria, namely the amount of masking in texture, the relative bilateral local density of the alignment, its internal regularity, and finally a redundancy reduction step. Extending tools of the *a contrario* detection theory, we show that all of these detection criteria can be naturally embedded in a single probabilistic *a contrario* model with a single user parameter, the number of false alarms. Our contribution to the *a contrario* theory is the use of sophisticated conditional events on random point sets, for which expectation we nevertheless find easy bounds. By these bounds the mathematical consistency of our detection model receives a simple proof. Our final algorithm also includes a new formulation of the *exclusion principle* in Gestalt theory to avoid redundant detections. The method is carefully compared to three state-of-the-art algorithms. Limitations of the final method are also illustrated and explained. In Chapter 4, this method will be successfully applied to real data in the classic problem of vanishing point detection.

The detection algorithm produced in this chapter can be tested online on any dot pattern at
`http://dev.ipol.im/~jirafa/ipol_demo/point_alignment_detection/`
(user: demo, password: demo)

2.1 Introduction

We will consider the problem of finding collinear subsets within a planar set of points. This problem arises in many contexts of data analysis: Alignments are among the simplest structures observable in a point set and 3D alignments are viewpoint-invariant structures. They constitute a classic example in statistical shape analysis [Small 1988]. Alignment detection is relevant in geology, where the alignment of features, for example earthquake epicenters, reflects underlying faults and joints [Lutz 1986; Hall et al. 2006; Hammer 2009]. In archaeology, geometric configurations of post holes, in particular alignments, often reveal the disposition of buildings even in presence of overlaps from different time periods [Small 1988; Litton and Restorick 1983; Broadbent 1980]. The computer vision applications include the detection of grids [Dubská et al. 2013], calibration

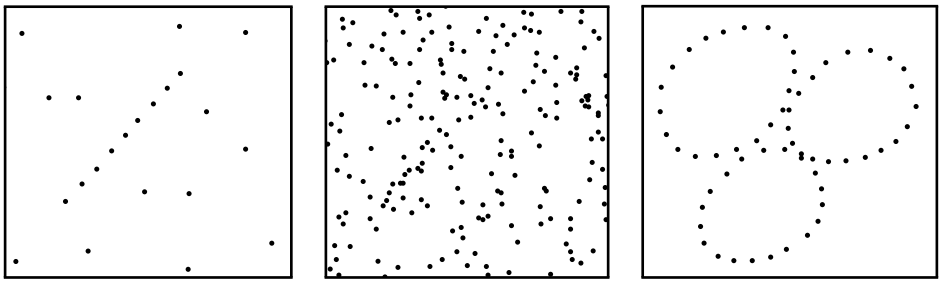


Figure 2.1: Illustration of the complexity of the point alignment problem: Exactly the same set of aligned dots is present in the three images, but it is only perceived as such in the first one. The second one is a classic "masking by texture" case and the third shows a "masking by structure", often called "Gestalt conflict".

patterns [Escalera and Armingol 2010] or vanishing points [Havel et al. 2013; Zhao et al. 2013], and the interpretation of high resolution remote sensing images [Vanegas et al. 2010].

Dot patterns are often used in the study of visual perception. Several psychophysical studies led by Uttal have investigated the effect of direction, quantity and spacing in dot alignment perception [Uttal et al. 1970; Uttal 1973, 1975, 1987]. The detection of collinear dots in noise was the target of other studies attempting to assess quantitatively the masking effect of the background noise [Smits et al. 1984; Kiryati et al. 1992; Tripathy et al. 1999; Mussap and Levi 2000]. The recent work of Preiss [2006] analyzes various perceptual tasks on dot patterns from a psychophysical and computational perspective. An interesting computational approach to detect gestalts in dot patterns is presented in Ahuja and Tuceryan [1989], although the study is limited to very regularly sampled patterns.

While it may seem that point alignments are simple structures, Figure 2.1 shows how complex an alignment event can be. From a purely factual point of view, the same alignment is present in the three figures. However, it is only perceived as such by most viewers in the first one. The second and the third figures illustrate two occurrences of the *masking phenomenon* discovered by gestaltists [Kanizsa 1991]: the *masking by texture*, which occurs when a geometric structure is surrounded by a clutter of randomly distributed similar objects or *distractors*, and the *masking by structure*, which happens when the structure is masked by other perceptually more relevant structures, a phenomenon also called *perceptual conflict* by gestaltists [Metzger 1975, 2006 (originally 1936); Kanizsa 1980]. The magic disappearance of the alignment in the second and third figures can be accounted for in two very different ways. For the first one, a probabilistic *a contrario* model [Desolneux et al. 2008] can lead to a quantitative prediction. The second one is explained by the winning intervention of three more powerful grouping laws, *good continuation*, *convexity* and *closure* in the perceptual conflict [Kanizsa 1979].

These examples show that a mathematical definition of point alignment perception is required before even starting to discuss how to detect them. A purely geometric-physical description is clearly not sufficient to account for the masking phenomenon. Indeed, an objective observer making use of a ruler would be able to state the existence of the very same alignment at the same precision on all three figures. But this statement would contradict our perception, as well as any reasonable computational (definition and) theory of alignment detection.

A classic approach to this problem uses the Hough transform [Hough Dec. 18, 1962; Duda and Hart 1972], first used for the detection of subatomic particles in bubble chamber pictures [Hough 1959]. To compute the Hough transform, each point votes in a parameter space for the lines that pass through it. After accumulation of the votes of all points, the lines that correspond to local

maxima in the parameter space are selected as detections. Several variations of the basic method were proposed; in particular, the methods proposed in Thrift and Dunn [1983]; Kiryati and M. [1991, 1992] are robust to errors in the point positions. When the Hough transform is applied to a random set of points, it will still find a local maximum which does not correspond to a significant collinear subset. A threshold on the number of votes is usually imposed to cope with this problem. Even if the Hough transform methods provide successful solutions in many applications [Wadge and Cross 1988], a sound criterion for setting this threshold is missing.

Other approaches use point clustering methods especially adapted to elongated clusters [Murtagh and Raftery 1984]. Using a particular distance between points and clusters [Davé 1989], general clustering algorithms can be used to detect collinear subsets [Frigui and Krishnapuram 1999; Figueiredo and Jain 2002]. The same problem can be approached using a parametric model fitting [Danuser and Stricker 1998; Fischler and Bolles 1981]. Given a parametric model, a criterion for point compatibility with the model, and a final validation criterion for the model, RANSAC [Fischler and Bolles 1981] is an efficient heuristic for fitting the searched model to the data. RANSAC, however, provides no solution on how to choose these criteria for a given model or application.

We are particularly interested in methods that provide an evaluation of the statistical significance of the detected aligned structures. An example in astronomy may illustrate the importance of such evaluation: In 1980 the discovery of several very precise alignments of quasars in the sky raised the question of a theory explaining this presence [Arp and Hazard 1980]. These alignments, however, were later dismissed by a statistical analysis, first by simulations [Edmunds and H. 1981] and then analytically [Zuiderwijk 1982], showing that alignments of such precision could easily occur just by chance.

The expected number of events where k among n random points are to be found in some rectangle of a given shape was already computed in 1950 using a Poisson random model [Mack 1950]. This could be the origin of the *strip* method for defining alignments as a large number of points covered by a thin rectangle (the thinner, the more precise). The same random model was used in Broadbent [1980], now explicitly used for detecting point alignments. But the alignment was defined differently: three points are considered aligned when the triangle formed by them is flat enough. Alignments of more points are evaluated by all the possible triangles observed among the points. Various theoretical results about the flat triangles methods are described in Kendall and Kendall [1980], where Poisson as well as Gaussian distributions are considered in different domain shapes.

Since then, many different algorithms have been proposed, most of them variations of the strip method. Monte Carlo simulations of random points provide the estimate of the significance in Zhang and Lutz [1989] while a binomial model is used in Amorese et al. [1999]. A set of heuristics are added in Arcasoya et al. [2004]. The method in Hall et al. [2006] also applies the strip method with a Poisson model, but the density is estimated locally. A refined statistical test, including angular statistics, is proposed in Hammer [2009]. Another approach combines a concatenation of center-surround operators with a meaningfulness evaluation [Lowe and Binford 1982].

Here we develop a method derived from the *a contrario* methodology proposed by Desolneux, Moisan and Morel [Desolneux et al. 2000, 2008]. It is a mathematical formalization of the *non-accidentalness principle* proposed for perception [Witkin and Tenenbaum 1983b; Albert and Hoffman 1995; Wagemans 1992] (sometimes called *Helmholtz principle*). In a nutshell, an observed structure is relevant if it would rarely occur by chance. This scheme has been repeatedly used in the past. In the words of David Lowe, "we need to determine the probability that each relation in the image could have arisen by accident, $P(a)$. Naturally, the smaller that this value is, the more likely the relation is to have a causal interpretation" [Lowe 1985, p.39]. The difference in the *a contrario*

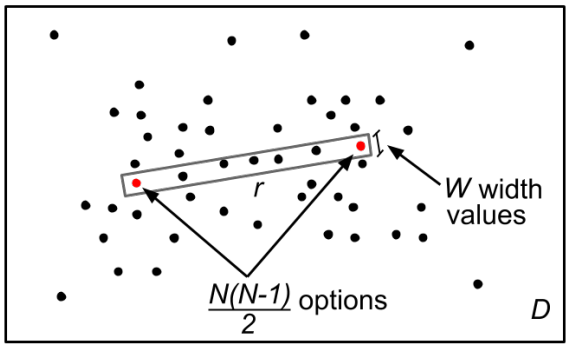


Figure 2.2: A schematic representation of the evaluated rectangle. In a domain with N points, there are $\frac{N(N-1)}{2}W$ possible rectangles. In this example, $N = 47$ and $k(r, \mathbf{x}) = 8$ among them are inside the rectangle r.

methodology is that the expectation of the number of false detections is controlled instead of the probability of observing a false detection. The resulting statistical framework provides estimates of significance similar in spirit to the methods mentioned before.

As a simple example to introduce the methodology, Desolneux et al. showed a point alignment detector using a simple strip method with a Poisson model [Desolneux et al. 2008, Section 3.2]. Even before, Kiryati et al. [1991] used a similar idea for setting thresholds in the voting space of the Hough transform. More recently, Tepper and Sapiro [2014a] proposed a novel approach for consensus problems based on combining bi-clustering and a contrario methods; one of their applications is in point alignment detection. Our goal in this work is to extend these initial methods into a working algorithm that can be used to successfully solve real image processing problems [Lezama et al. 2014a]. To cope with obvious objections and counterexamples, we shall prove that three new features are necessary to handle the variety of alignments. We shall show that a reliable algorithm requires: a) a *local* Poisson density estimation, b) an evaluation of the *regularity of the spacing* of the points in the alignment, and c) a criterion to *select the best interpretation among redundant detections*. This work concentrates on the criteria for obtaining the best possible result with an unsupervised algorithm, neglecting the efficiency concerns, which remains as a future line of research.

The rest of this chapter is organized as follows: Section 2.2 introduces the basic concepts and techniques used in the state-of-the-art point alignment detectors, and describes the classic strip method. Sections 2.3, 2.4 and 2.5 improve this basic method by incorporating local point density estimation, lateral estimation, and measurements of the regularity of the point spacing. Section 2.6 discusses how to cope with the redundancy of detections. Section 2.7 analyses the complexity of the final algorithm, and a possible acceleration is proposed in Section 2.8. Section 2.9 shows experiments and comparisons. Section 2.10 concludes the study.

2.2 Basic Point Alignment Detector

Consider a set of N points defined in a domain D with area S_D, see Figure 2.2. We are interested in detecting groups of points that are well aligned. A reasonable *a contrario* hypothesis H_0 for this problem is to suppose that the N points are the result of a random process where points are independent and uniformly distributed in the domain. Recently, a technique for relaxing the independence assumption in the *a contrario* framework has been presented [Myaskouvskey et al.

2013], showing more accurate predictions of the false detections rate. In this work however, since the basic elements are not computed from image pixels we will keep the independence assumption. Citing Lowe again, "One of the most general and obvious assumptions we can make is to assume a background of independently positioned objects in three-space, which in turn implies independently positioned projections of the objects in the image." [Lowe 1985, p.39] This does not mean that the method will only work when the background points follow exactly this hypothesis. What is important is that this is a good model for isotropic elements where any alignment is accidental. As we will see in practice, the method discriminates well between accidental alignments and causal ones. The question is then to evaluate whether the presence of aligned points contradicts the *a contrario* model or not.

Given an observed set of N points $\mathbf{x} = \{x_i\}_{i=1...N}$ and a rectangle r (a candidate for alignment), we will denote by $k(r, \mathbf{x})$ the number of those points observed inside r. The decision of whether to keep this candidate or not is based on two principles: a good candidate should be non-accidental, and any equivalent or better candidate should be kept as well. The degree of non-accidentalness of a rectangle r can therefore be measured by how small the probability $\mathbb{P}\big[k(r, \mathbf{X}) \geq k(r, \mathbf{x})\big]$ is, where $\mathbf{X}$ denotes a random set of N points following H_0. In the same vein, a rectangle r' will be considered at least as good as r given the observation $\mathbf{x}$, if $\mathbb{P}\big[k(r', \mathbf{X}) \geq k(r', \mathbf{x})\big] \leq \mathbb{P}\big[k(r, \mathbf{X}) \geq k(r, \mathbf{x})\big]$.

The question is how to control the expected number of accidental detections [Desolneux et al. 2008]. Given that N_{tests} candidates will be tested, the expected number of rectangles which are as good as r under H_0 is less than

$$N_{tests} \cdot \mathbb{P}\Big[k(r, \mathbf{X}) \geq k(r, \mathbf{x})\Big]. \tag{2.1}$$

The H_0 stochastic model fixes the probability law of the random number of points in the rectangle, $k(r, \mathbf{X})$. The discrete nature of this law implies that (2.1) is not actually the expected value but an upper bound of it [Desolneux et al. 2008; Grompone von Gioi and Jakubowicz 2009]. Let us now analyze the two factors in (2.1).

Under the *a contrario* hypothesis H_0 (a planar Poisson process [Miles 1970]), the probability that one point falls into the rectangle r is $p = \frac{S_r}{S_D}$, where S_r is the area of the rectangle and S_D the area of the domain. As a consequence of the independence of the random points, $k(r, \mathbf{X})$ follows a binomial distribution. Thus, the probability term $\mathbb{P}\big[k(r, \mathbf{X}) \geq k(r, \mathbf{x})\big]$ is given by

$$\mathbb{P}\Big[k(r, \mathbf{X}) \geq k(r, \mathbf{x})\Big] = \mathcal{B}\big(N, k(r, \mathbf{x}), p\big) \tag{2.2}$$

where $\mathcal{B}(n, k, p)$ is the tail of the binomial distribution

$$\mathcal{B}(n, k, p) = \sum_{j=k}^{n} \binom{n}{j} p^j (1-p)^{n-j}. \tag{2.3}$$

The *number of tests* N_{tests} corresponds to the total number of rectangles that could contain an alignment, which in turn is proportional to the number of pairs of points defining such rectangles. With a set of N points this gives $\frac{N(N-1)}{2}$ different pairs of points. The set of rectangle widths to be tested must be specified *a priori* as well. In the *a contrario* approach, a compromise must be found between the number of tests and the precision of the structures that are being sought for. The larger the number of tests, the lower the statistical relevance of detections, but also the more precise. However, if the set of tests is chosen wisely, structures fitting accurately the tests will have a very low probability of occurrence under H_0 and will therefore be more significant.

Algorithm 1: Basic point alignment detector

input : A set $\mathbf{x}$ of N points $[W = 8, \varepsilon = 1]$
output: A list $\mathbf{out}$ of point alignments

```
for i = 1 to N do
    for j = 1 to i − 1 do
        l ← distance(x_i, x_j)
        w ← l/10
        for 1 to W do
            r ← rect(x_i, x_j, w)
            Compute NFA_1(r, x) [Eq. (5)]
            if NFA_1(r, x) ≤ ε then
                out ← r
            w ← w/√2
```

For a particular problem, one may have reasons to restrict the shape of rectangles. Nevertheless, this inquiry is not aimed at any particular application. Thus, we will rely on the following general criteria. An alignment should be an elongated structure, so a minimal ratio between the length and the width of the rectangle must be fixed. Then, a fixed number of widths must be tested, decreasing geometrically from the maximal width. (The choice of a geometric series is justified by the obvious scale invariance of the detection problem.) Our implementation uses a $\mathrm{length}/\mathrm{width}_{max}$ ratio of 10 and a geometric series of 8 width values with a factor $1/\sqrt{2}$. The total number of widths to be tested will be denoted by W. Then the total number of tested rectangles is

$$N_{tests} = \frac{N(N-1)}{2} W. \tag{2.4}$$

The fundamental quantity of an *a contrario* approach is the Number of False Alarms (NFA) associated with a rectangle r and a set of points $\mathbf{x}$,

$$\begin{aligned} \mathrm{NFA}_1(r, \mathbf{x}) &= N_{tests} \cdot \mathbb{P}\Big[k(r, \mathbf{X}) \geq k(r, \mathbf{x})\Big] \\ &= \frac{N(N-1)}{2} W \cdot \mathcal{B}\Big(N, k(r, \mathbf{x}), p\Big). \end{aligned} \tag{2.5}$$

This quantity gives a precise meaning to Eq. (2.1). It will be interpreted as a bound of the expected number of rectangles containing enough *points* to be as rare as r under H_0. When the NFA associated with a rectangle is large, this means that such an event is to be expected under the *a contrario* model and therefore is not relevant. On the other hand, when the NFA is small, the event is rare and probably meaningful. A rarity threshold ε must nevertheless be fixed for each application. Rectangles with $\mathrm{NFA}_1(r, \mathbf{x}) \leq \varepsilon$ will be called *ε-meaningful rectangles* [Desolneux et al. 2008], constituting the detection result of the algorithm. A pseudo-code of this method is described in Algorithm 1.

Theorem 1 (Desolneux et al. [2008])**.**

$$\mathbb{E}\left[\sum_{r \in \mathcal{R}} \mathbb{1}_{\mathrm{NFA}_1(r, \mathbf{X}) \leq \varepsilon}\right] \leq \varepsilon$$

where $\mathbb{E}$ *is the expectation operator,* $\mathbb{1}$ *is the indicator function,* $\mathcal{R}$ *is the set of test rectangles, and* $\mathbf{X}$ *is a random set of points under* H_0*.*

The theorem states that the average number of ε*-meaningful rectangles* under the *a contrario* model H_0 is bounded by ε. Thus, the number of detections in noise is controlled by ε and it can be made as small as desired. In other words, this detector satisfies the non-accidentalness principle.

Proof. We define $\hat{k}(r)$ as

$$\hat{k}(r) = \min\left\{\kappa \in \mathbb{N},\ \mathbb{P}\big[k(r,\mathbf{X}) \geq \kappa\big] \leq \frac{\varepsilon}{\frac{N(N-1)}{2}W}\right\}.$$

Then, $\mathrm{NFA}_1(r,\mathbf{X}) \leq \varepsilon$ is equivalent to $k(r,\mathbf{X}) \geq \hat{k}(r)$. Now,

$$\begin{aligned}\mathbb{E}\left[\sum_{r\in\mathcal{R}} \mathbb{1}_{\mathrm{NFA}_1(r,\mathbf{X})\leq\varepsilon}\right] &= \sum_{r\in\mathcal{R}} \mathbb{P}\big[\mathrm{NFA}_1(r,\mathbf{X}) \leq \varepsilon\big] \\ &= \sum_{r\in\mathcal{R}} \mathbb{P}\left[k(r,\mathbf{X}) \geq \hat{k}(r)\right].\end{aligned}$$

But, by definition of $\hat{k}(r)$ we know that

$$\mathbb{P}\left[k(r,\mathbf{X}) \geq \hat{k}(r)\right] \leq \frac{\varepsilon}{\frac{N(N-1)}{2}W}.$$

and using that $\#\mathcal{R} = \frac{N(N-1)}{2}W$ we get

$$\mathbb{E}\left[\sum_{r\in\mathcal{R}} \mathbb{1}_{\mathrm{NFA}_1(r,\mathbf{X})\leq\varepsilon}\right] \leq \sum_{r\in\mathcal{R}} \frac{\varepsilon}{\frac{N(N-1)}{2}W} = \varepsilon$$

which concludes the proof. □

As shown in Desolneux et al. [2008], the detection result is not very sensitive to the value of ε. Following Desolneux et al. [2000, 2008], we shall therefore fix $\varepsilon = 1$ for our experiments. This corresponds to accepting on average at most one false detection per data set in the *a contrario* model.

Figure 2.3 shows the results of the basic algorithm in two simple cases. The results are as expected: the visible alignment in the first example is detected, while no detection is produced in the second. Actually, the points in the first example are also present in the second one, but the addition of random points masks the alignment to our perception. The first example produces many redundant detections; this issue will be addressed in Section 2.6.

2.3 Local Density Estimation

The basic point alignment detector of section 2.2 takes as *a contrario* assumption a uniform point density in the whole domain and evaluates alignments as a local excess with respect to this global density. This comparison is nevertheless too restrictive, because these alignments have been detected as *local violations* of a global uniformity. Consider instead a configuration of points with two zones of different point density, like in Figure 2.4 (a). Applying the basic alignment detector

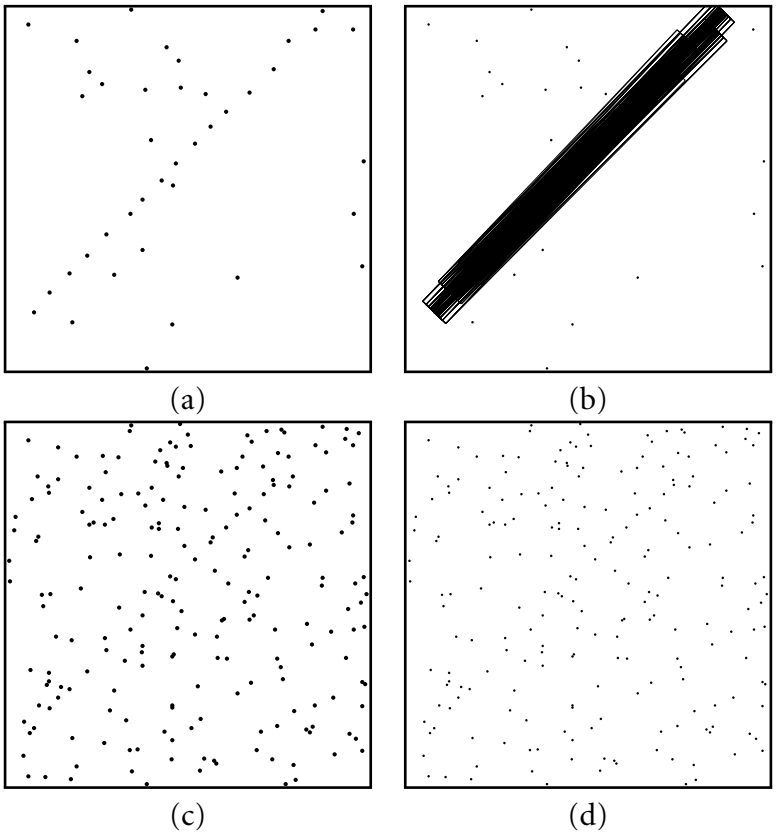


Figure 2.3: Results from the basic point alignment detector (Algorithm 1). **(a)** and **(c)** are the input data, and **(b)** and **(d)** are the corresponding results. Each detection is represented by a rectangle. In **(b)** the algorithm correctly detects the obvious alignment. Notice that multiple and redundant rectangles were detected; this issue will be dealt with in Section 2.6. The data set **(c)** contains the same set of points in **(a)** plus added noise points. The aligned points are still present but hardly perceptible. The algorithm handles correctly this masking phenomenon and produces no detection.

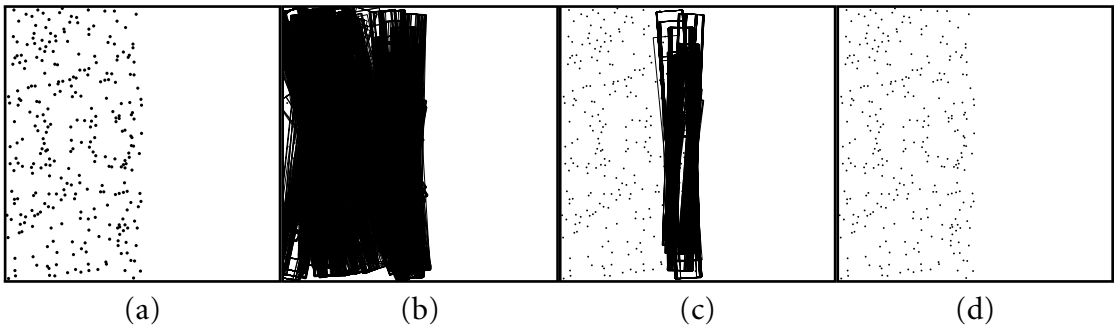


Figure 2.4: Local vs. global density estimation. **(a)** The set of points. **(b)** Alignments found using global density estimation (Algorithm 1). The many detected rectangles indeed have a high point density compared to the average image density used as background model. **(c)** Alignments found using local density estimation (Algorithm 2). The local density is lower on the border, hence the deceptive detection. **(d)** No alignment is found when the local density is estimated by the maximum density on both sides of the alignment (Algorithm 3).

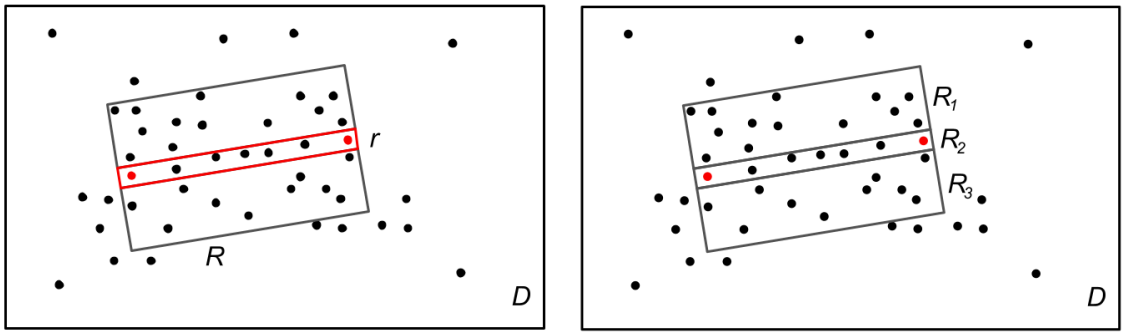


Figure 2.5: **Left:** The point density is estimated in the local window R surrounding the alignment r. **Right:** In a refined version of the algorithm, the density of points is measured on each side of the evaluated rectangle. The maximum of the densities in R_1 and R_3 is taken as an estimation of the point density in both R_1 and R_3.

yields an unexpected detection shown in Figure 2.4 (b). Each of the detected rectangles certainly has a non-accidental excess of points in the rectangles with respect to the global density, but this is definitely not what we are looking for. This example shows that we are actually interested in non-accidental events with an excess of points conditioned to the observation of a *local density* (which may well be lower or higher than the global density). Such local density estimations for the random point models have been used in Amorese et al. [1999]; Hall et al. [2006]; Hammer [2009]. In the interpretation proposed here, a more sophisticated definition of the alignment event should not measure the non-accidentalness by an unusually small probability, but by an unusually small *conditional* probability.

The local density is estimated by counting the points in a rectangular local window, with the same length as the alignment and a given width. To account for the scale invariance of the detection, the width of the local window is proportional to the length of the alignment. For every alignment, a number of proportion ratios or scales are tried. The largest window is square of side equal to the length of the alignment. Then a fixed number $L = 8$ of widths in geometric series are tried. The choice for a geometric series with factor $1/\sqrt{2}$ is again justified by the scale invariance of the detection problem. The *number of tests* N_{tests} corresponds to the total number of observations performed, which in turn is related to the number of rectangles and the different local windows evaluated for each rectangle. For N points and L different sizes of local windows, this gives $\frac{N(N-1)}{2}WL$ different tests.

When the rectangle to be tested lies near the border of the domain, the local window may be partly outside it, where no point information is available, leading to a wrong density estimation. This also happens when the rectangle covers the diagonal of the domain. A symmetric extension of the point set across the domain boundary is used to estimate the point density in windows meeting the outside. The candidates are still selected among the original points.

Let R be the local window surrounding the alignment r, as shown in Figure 2.5 (left). The probability of one point in R falling in r is $p = \frac{S_r}{S_R}$ where S_r and S_R are the areas of r and R respectively. The degree of non-accidentalness of an observation will be measured by the probability that a rectangle has a higher density than its surroundings, conditioned by the observation of the

surrounding density. The NFA for the new detector is accordingly defined as

$$\begin{aligned}\mathrm{NFA}_2(r,R,\mathbf{x}) &= N_{tests}\cdot\mathbb{P}\Big[k(r,\mathbf{X})\geq k(r,\mathbf{x})\,\Big|\,n(R,\mathbf{X})=n(R,\mathbf{x})\Big]\\ &= \frac{N(N-1)}{2}WL\cdot\mathcal{B}\Big(n(R,\mathbf{x}),k(r,\mathbf{x}),p\Big),\end{aligned}\tag{2.6}$$

where $n(R,\mathbf{x})$ is the number of points observed in $\mathbf{x}$ inside R. The pseudo-code for this method is described in Algorithm 2.

Theorem 2.

$$\mathbb{E}\left[\sum_{r\in\mathcal{R}}\sum_{R\in\mathcal{R}'(r)}\mathbb{1}_{\mathrm{NFA}_2(r,R,\mathbf{X})\leq\varepsilon}\right]\leq\varepsilon$$

where $\mathbb{E}$ is the expectation operator, $\mathbb{1}$ is the indicator function, $\mathcal{R}$ is the set of rectangles considered, $\mathcal{R}'(r)$ is the set of surrounding local windows for each rectangle r, and $\mathbf{X}$ is a random set of points on H_0.

Proof. We define $\hat{k}(r,R,M)$ as

$$\hat{k}(r,R,M) \;=\; \min\left\{\kappa\;\in\;\mathbb{N},\mathbb{P}\big[k(r,\mathbf{X})\;\geq\;\kappa\,\big|\,n(R,\mathbf{X})\;=\;M\big]\;\leq\;\frac{\varepsilon}{\frac{N(N-1)}{2}W\Omega}\right\}.$$

Here, R determines the domain of the local window and M the number of points in it. Notice that the probabilistic model inside R conditioned to the fact that the number of observed points is M, is still uniform and independent. Thus, the conditional law of the number of points inside any subset of R still follows a binomial law. Thus, given that the number of points observed in R is M, $\mathrm{NFA}_2(r,R,\mathbf{X})\leq\varepsilon$ is equivalent to $k(r,\mathbf{X})\geq\hat{k}(r,R,M)$. Now,

$$\mathbb{E}\left[\sum_{r\in\mathcal{R}}\sum_{R\in\mathcal{R}'(r)}\mathbb{1}_{\mathrm{NFA}_2(r,R,\mathbf{X})\leq\varepsilon}\right]=\sum_{r\in\mathcal{R}}\sum_{R\in\mathcal{R}'(r)}\mathbb{P}\big[\mathrm{NFA}_2(r,R,\mathbf{X})\leq\varepsilon\big]=$$

$$\sum_{r\in\mathcal{R}}\sum_{R\in\mathcal{R}'(r)}\sum_{M=0}^{N}\mathbb{P}\big[\mathrm{NFA}_2(r,R,\mathbf{X})\leq\varepsilon\,\big|\,n(R,\mathbf{X})=M\big]\cdot\mathbb{P}\big[n(R,\mathbf{X})=M\big]=$$

$$\sum_{r\in\mathcal{R}}\sum_{R\in\mathcal{R}'(r)}\sum_{M=0}^{N}\mathbb{P}\Big[k(r,\mathbf{X})\geq\hat{k}(r,R,M)\,\Big|\,n(R,\mathbf{X})=M\Big]\cdot\mathbb{P}\big[n(R,\mathbf{X})=M\big].$$

But, by definition of $\hat{k}(r,R,M)$ we know that

$$\mathbb{P}\Big[k(r,\mathbf{X})\geq\hat{k}(r,R,M)\,\Big|\,n(R,\mathbf{X})=M\Big]\leq\frac{\varepsilon}{\frac{N(N-1)}{2}W\Omega},$$

and using that $\#\mathcal{R}=\frac{N(N-1)}{2}W$, $\#\mathcal{R}'(r)=\Omega$ and $\sum_{M=0}^{N}\mathbb{P}\big[n(R,\mathbf{X})=M\big]=1$ we get

$$\mathbb{E}\left[\sum_{r\in\mathcal{R}}\sum_{R\in\mathcal{R}'(r)}\mathbb{1}_{\mathrm{NFA}_2(r,R,\mathbf{X})\leq\varepsilon}\right]\leq\sum_{r\in\mathcal{R}}\sum_{R\in\mathcal{R}'(r)}\frac{\varepsilon}{\frac{N(N-1)}{2}W\Omega}\sum_{M=0}^{N}\mathbb{P}\big[n(R,\mathbf{X})=M\big]=\varepsilon$$

which concludes the proof. □

Algorithm 2: Local density alignment detector

input : A set $\mathbf{x}$ of N points $[W = 8, L = 8, \varepsilon = 1]$
output: A list $\mathbf{out}$ of point alignments

```
for i = 1 to N do
    for j = 1 to i − 1 do
        l ← distance(x_i, x_j)
        w ← l/10
        for 1 to W do
            r ← rect(x_i, x_j, w)
            w_L ← l
            for 1 to L do
                R ← rect(x_i, x_j, w_L)
                Compute NFA_2(r, R, x) [Eq. (6)]
                if NFA_2(r, R, x) ≤ ε then
                    out ← r
                w_L ← w_L/√2
            w ← w/√2
```

2.4 Lateral Density Estimation

While the local density estimation can provide a more adjusted background model, it can also introduce new problems such as a "border effect", as shown in Figure 2.4 (c). Indeed, the density estimation is lower on the border of the left half of the image than inside it. Thus, the previous algorithm (Algorithm 2) detects alignments on the border with non-accidental, meaningful excess with respect to the local density.

In order to avoid this effect, the more sophisticated Algorithm 3 used in Figure 2.4 (d) takes, as a conservative estimation of the background density, the *maximum of the densities* measured on both sides of the alignment. In short, to be detected, an alignment must show a higher point density than in both regions immediately on its left and right. This local alignment detector is therefore similar to a classic second order Gabor filter where an elongated expiatory region is surrounded by two inhibitory regions. The local density estimation is calculated as illustrated in Figure 2.5 (right): The local window is divided in three parts. R_1 is the rectangle formed by the area of the local window on the left of the alignment. R_3 is the area of the local window on the right of the alignment, and R_2 is the rectangle which forms the candidate alignment. Next, the algorithm counts the numbers of points M_1, M_2, and M_3 in R_1, R_2 and R_3, respectively, and defines the conservative estimate of the local number of points as

$$n^*(R, \mathbf{x}) = 2 \max(M_1, M_3) + M_2. \tag{2.7}$$

We then define the NFA of the event "the density in R_2 has a significant excess with respect to the

density estimated in R" by

$$\begin{aligned}\mathrm{NFA}_3(r,R,\mathbf{x}) &= N_{tests}\cdot\mathbb{P}\Big[k(r,\mathbf{X})\geq k(r,\mathbf{x})\,\Big|\,n(R,\mathbf{X})=n^*(R,\mathbf{x})\Big]\\ &= \frac{N(N-1)}{2}WL\cdot\mathcal{B}\Big(n^*(R,\mathbf{x}),k(r,\mathbf{x}),p\Big).\end{aligned}\tag{2.8}$$

Indeed, conditioned to the fact that we assume $n(R,\mathbf{X}) = n^*(R,\mathbf{x})$ under the model H_0, the $n^*(R,\mathbf{x})$ points in R are still uniformly and independently distributed. The pseudo-code for this method is described in Algorithm 3.

Theorem 3.

$$\mathbb{E}\left[\sum_{r\in\mathcal{R}}\sum_{R\in\mathcal{R}'(r)}\mathbb{1}_{\mathrm{NFA}_3(r,R,\mathbf{X})\leq\varepsilon}\right]\leq\varepsilon$$

where $\mathbb{E}$ is the expectation operator, $\mathbb{1}$ is the indicator function, $\mathcal{R}$ is the set of rectangles considered, $\mathcal{R}'(r)$ is the set of surrounding local windows for each rectangle r, and $\mathbf{X}$ is a random set of points on H_0.

Proof. First define $\hat{k}(r,R,M)$ by

$$\hat{k}(r,R,M) = \min\left\{\kappa\in\mathbb{N},\mathbb{P}\big[k(r,\mathbf{X})\geq\kappa\,\big|\,n(R,\mathbf{X})=M\big]\leq\frac{\varepsilon}{\frac{N(N-1)}{2}W\Omega}\right\}.$$

When $n(R,\mathbf{X})=M$, we have that $\mathrm{NFA}_3(r,R,\mathbf{X})\leq\varepsilon$ is equivalent to $k(r,\mathbf{X})\geq\hat{k}(r,R,M)$. In consequence,

$$\mathbb{E}\left[\sum_{r\in\mathcal{R}}\sum_{R\in\mathcal{R}'(r)}\mathbb{1}_{\mathrm{NFA}_3(r,R,\mathbf{X})\leq\varepsilon}\right]=\sum_{r\in\mathcal{R}}\sum_{R\in\mathcal{R}'(r)}\mathbb{P}\big[\mathrm{NFA}_3(r,R,\mathbf{X})\leq\varepsilon\big]=$$

$$\sum_{r\in\mathcal{R}}\sum_{R\in\mathcal{R}'(r)}\sum_{M=0}^{2N}\mathbb{P}\big[\mathrm{NFA}_3(r,R,\mathbf{X})\leq\varepsilon\,\big|\,n(R,\mathbf{X})=M\big]\cdot\mathbb{P}\big[n(R,\mathbf{X})=M\big]=$$

$$\sum_{r\in\mathcal{R}}\sum_{R\in\mathcal{R}'(r)}\sum_{M=0}^{2N}\mathbb{P}\left[k(r,\mathbf{X})\geq\hat{k}(r,R,M)\,\Big|\,n(R,\mathbf{X})=M\right]\cdot\mathbb{P}\big[n(R,\mathbf{X})=M\big].$$

Note that, because of the maximum density estimation, the "estimated" number of points inside a rectangle can theoretically be as large as $2N$. But, by definition of $\hat{k}(r,R,M)$,

$$\mathbb{P}\left[k(r,\mathbf{X})\geq\hat{k}(r,R,M)\,\Big|\,n(R,\mathbf{X})=M\right]\leq\frac{\varepsilon}{\frac{N(N-1)}{2}W\Omega},$$

and using that $\#\mathcal{R}=\frac{N(N-1)}{2}W$, $\#\mathcal{R}'(r)=\Omega$ and $\sum_{M=0}^{2N}\mathbb{P}\big[n(R,\mathbf{X})=M\big]=1$ we get

$$\mathbb{E}\left[\sum_{r\in\mathcal{R}}\sum_{R\in\mathcal{R}'(r)}\mathbb{1}_{\mathrm{NFA}_3(r,R,\mathbf{X})\leq\varepsilon}\right]\leq\sum_{r\in\mathcal{R}}\sum_{R\in\mathcal{R}'(r)}\frac{\varepsilon}{\frac{N(N-1)}{2}W\Omega}\sum_{M=0}^{2N}\mathbb{P}\big[n(R,\mathbf{X})=M\big]=\varepsilon,$$

which concludes the proof. □

Algorithm 3: Lateral local alignment detector

input : A set $\mathbf{x}$ of N points [$W = 8$, $L = 8$, $\varepsilon = 1$]
output: A list $\mathbf{out}$ of point alignments

```
1  for i = 1 to N do
2      for j = 1 to i − 1 do
3          l ← distance(x_i, x_j)
4          w ← l/10
5          for 1 to W do
6              r ← rect(x_i, x_j, w)
7              w_L ← l
8              for 1 to L do
9                  R_1 ← local-win-left(x_i, x_j, w_L)
10                 R_3 ← local-win-right(x_i, x_j, w_L)
11                 Compute NFA_3(r, R, x) [Eq. (8)]
12                 if NFA_3(r, R, x) ≤ ε then
13                     out ← r
14                 w_L ← w_L/√2
15             w ← w/√2
```

2.5 Alignment Regularity

There is still an objection to Algorithm 3: One can stir wrong detections by introducing small point clusters as shown in Figure 2.6 (left). The detected alignment in Figure 2.6 (center) seems clearly wrong. It is nevertheless explainable in the setting of Algorithm 3: there is indeed a meaningful point density excess inside the red rectangle. But this excess is caused by the clusters, not by what could be termed an alignment. While the algorithm counted every point, human perception seems to group the small clusters into a single entity, and to count them only once. This unwanted result is a consequence of the fact that Algorithm 3 is searching for elongated clusters of higher density without any cluster regularity requirement. As suggested in other studies [Preiss 2006; Tripathy et al. 1999; Uttal 1973], the density is not the only property that makes an alignment perceptually meaningful; another characteristic to consider is the uniform spacing or regularity of the points in it, which the gestaltists call the law of *constant spacing*. To cope with both issues (avoiding small clusters and favoring regular spacing) a more advanced version of the alignment detector divides each candidate rectangle into equal *boxes*. Instead of counting the total number of points, the algorithm counts the number of boxes that are occupied by at least one point. We call them *occupied* boxes. In this way, the minimal NFA is attained when the points are perfectly distributed along the alignment. In addition, a concentrated cluster in the alignment has no more influence on the alignment detection than a single point in the same position.

We want to estimate the expected number of occupied boxes in the background model H_0. The probability of one point falling in one of the boxes is $p_0 = \frac{S_B}{S_L}$, where S_B and S_L are the areas of the boxes and the local window respectively. Then, the probability of having one box occupied

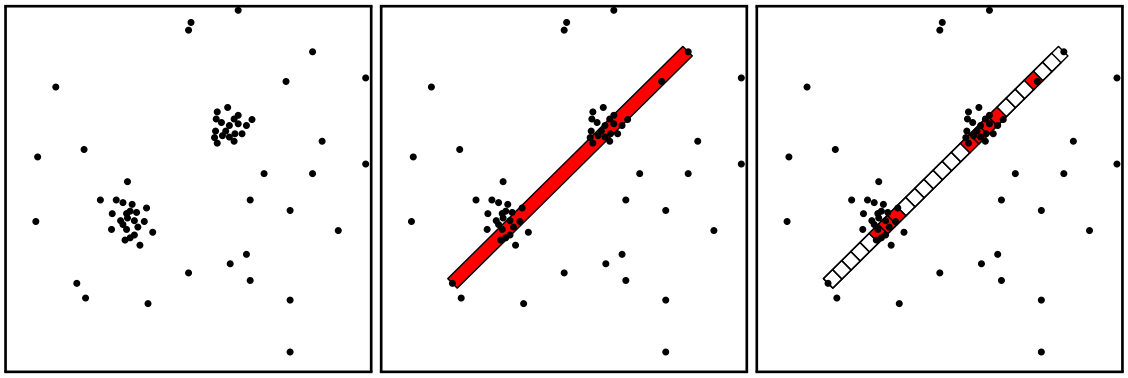


Figure 2.6: **Left:** Dot pattern with two point clusters but no alignment. **Center:** A thin rectangle with a high point density was found, but it would constitute a false detection. **Right:** The algorithm divides the rectangle into boxes and counts the occupied ones, avoiding this misleading cluster effect. The occupied boxes are marked in red. In this case, no alignment is detected.

by at least one of the $n^*(R, \mathbf{x})$ points (i.e., of an *occupied* box) is

$$p_1(R, c) = 1 - (1 - p_0)^{n^*(R,\mathbf{x})}. \tag{2.9}$$

We will denote by $b(r, c, \mathbf{x})$ the observed number of occupied boxes in the rectangle r when divided into c boxes. Finally, the probability of having at least $b(r, c, \mathbf{x})$ of the c boxes occupied is

$$\mathcal{B}\big(c, b(r, c, \mathbf{x}), p_1(R, c)\big). \tag{2.10}$$

A set $\mathcal{C}$ of different values are tried for the number of boxes c into which the rectangle is divided, and the one producing the lowest NFA is taken. Thus, the number of tests must be multiplied by its cardinality $\#\mathcal{C} = C$. In practice we set $C = \sqrt{N}$ and that leads to

$$N_{tests} = \frac{N(N-1)}{2} WLC = \frac{N(N-1)}{2} WL\sqrt{N}. \tag{2.11}$$

The NFA of the new event definition is then

$$\begin{aligned} \mathrm{NFA}_4(r, R, c, \mathbf{x}) &= N_{tests} \cdot \mathbb{P}\Big[b(r, c, \mathbf{X}) \geq b(r, c, \mathbf{x}) \,\Big|\, n(R, \mathbf{X}) = n^*(R, \mathbf{x})\Big] \\ &= \frac{N(N-1)}{2} WLC \cdot \mathcal{B}\big(c, b(r, c, \mathbf{x}), p_1(R, c)\big). \end{aligned} \tag{2.12}$$

Figure 2.6 (right) shows an example of the resulting algorithm and we will show more in section 2.9. Algorithm 4 presents the pseudo-code for this final refined version of the alignment detector.

Theorem 4.

$$\mathbb{E}\left[\sum_{r\in\mathcal{R}} \sum_{R\in\mathcal{R}'(r)} \sum_{c\in\mathcal{C}} \mathbb{1}_{\mathrm{NFA}_4(r,R,c,\mathbf{X})\leq\varepsilon}\right] \leq \varepsilon$$

where $\mathbb{E}$ is the expectation operator, $\mathbb{1}$ is the indicator function, $\mathcal{R}$ is the set of rectangles considered, $\mathcal{R}'(r)$ is the set of surrounding local windows for each rectangle r, $\mathcal{C}$ is the set of number of boxes tested, and $\mathbf{X}$ is a random set of points under H_0.

Proof. We define $\hat{b}(r, R, c, M)$ as

$$\hat{b}(r, R, c, M) \;=\; \min\left\{\beta \;\in\; \mathbb{N}, \mathbb{P}\big[b(r, c, \mathbf{X}) \;\geq\; \beta \,\big|\, n(R, \mathbf{X}) \;=\; M\big] \;\leq\; \frac{\varepsilon}{\frac{N(N-1)}{2} WLC}\right\}.$$

R determines the domain of the local window and M the number of points in it. The probabilistic model inside R, conditioned to the fact that the number of observed points is M, is still uniform and independent, and the conditional law of the number of points inside any subset of R follows a binomial law. Then, $\text{NFA}_4(r, R, c, \mathbf{X}) \leq \varepsilon$ is equivalent to $b(r, c, \mathbf{X}) \geq \hat{b}(r, R, c, M)$ when $M = n^*(R, \mathbf{X})$. Now,

$$\mathbb{E}\left[\sum_{r\in\mathcal{R}}\sum_{R\in\mathcal{R}'(r)}\sum_{c\in\mathcal{C}} \mathbb{1}_{\text{NFA}_4(r,R,c,\mathbf{X})\leq\varepsilon}\right] = \sum_{r\in\mathcal{R}}\sum_{R\in\mathcal{R}'(r)}\sum_{c\in\mathcal{C}} \mathbb{P}\big[\text{NFA}_4(r, R, c, \mathbf{X}) \leq \varepsilon\big] =$$

$$\sum_{r\in\mathcal{R}}\sum_{R\in\mathcal{R}'(r)}\sum_{c\in\mathcal{C}}\sum_{M=0}^{2N} \mathbb{P}\big[\text{NFA}_4(r, R, c, \mathbf{X}) \leq \varepsilon \,\big|\, n(R, \mathbf{X}) = M\big] \cdot \mathbb{P}\big[n(R, \mathbf{X}) = M\big] =$$

$$\sum_{r\in\mathcal{R}}\sum_{R\in\mathcal{R}'(r)}\sum_{c\in\mathcal{C}}\sum_{M=0}^{2N} \mathbb{P}\big[b(r, c, \mathbf{X}) \geq \hat{b}(r, R, c, M) \,\big|\, n(R, \mathbf{X}) = M\big] \cdot \mathbb{P}\big[n(R, \mathbf{X}) = M\big]. \quad (2.13)$$

Note that, because of the maximum density estimation $n^*(R, \mathbf{x})$, the estimated number of points inside a rectangle can theoretically be as large as $2N$, and thus the range for M. By definition of $\hat{b}(r, R, c, M)$,

$$\mathbb{P}\left[b(r, c, \mathbf{X}) \geq \hat{b}(r, R, c, M) \,\big|\, n(R, \mathbf{X}) = M\right] \leq \frac{\varepsilon}{\frac{N(N-1)}{2} WLC},$$

and using $\#\mathcal{R} = \frac{N(N-1)}{2} W$, $\#\mathcal{R}'(r) = L$, $\#\mathcal{C} = C$ and $\sum_{M=0}^{2N} \mathbb{P}\big[n(R, \mathbf{X}) = M\big] = 1$, we get

$$\mathbb{E}\left[\sum_{r\in\mathcal{R}}\sum_{R\in\mathcal{R}'(r)}\sum_{c\in\mathcal{C}} \mathbb{1}_{\text{NFA}_4(r,R,c,\mathbf{X})\leq\varepsilon}\right] \leq \sum_{r\in\mathcal{R}}\sum_{R\in\mathcal{R}'(r)}\sum_{c\in\mathcal{C}} \frac{\varepsilon}{\frac{N(N-1)}{2} WLC} \sum_{M=0}^{2N} \mathbb{P}\big[n(R, \mathbf{X}) = M\big] = \varepsilon,$$

which concludes the proof. □

2.6 Redundancy

As was observed in Figure 2.3, all the described alignment detectors may produce redundant detections. Given a very meaningful alignment, many smaller or larger rectangles overlapping the main alignment are also meaningful. This redundancy phenomenon can involve points that belong to the real alignment as well as background points near the alignment, as illustrated in Figure 2.7. The question is how to detect the best rectangle, both explaining and masking the redundant detections.

A classic redundancy elimination method is Canny's non-maximal suppression Canny [1986], where maximal edge points inhibits their neighbors. A non-maximal suppression is also needed when using the Hough transform [Duda and Hart 1972]. Gerig and Klein [1986] and Gerig [1987]

Algorithm 4: Point alignment detector with boxes

input : A set $\mathbf{x}$ of N points $[W = 8, L = 8, \varepsilon = 1]$
output: A list $\mathbf{out}$ of point alignments

```
for i = 1 to N do
    for j = 1 to i − 1 do
        l ← distance(x_i, x_j)
        w ← l/10
        for 1 to W do
            r ← rect(x_i, x_j, w)
            w_L ← l
            for 1 to L do
                R_1 ← local-win-left(x_i, x_j, w_L)
                R_3 ← local-win-right(x_i, x_j, w_L)
                for c ∈ C do
                    Compute NFA_4(r, R, c, x) [eq.2.12]
                    if NFA_4(r, R, c, x) ≤ ε then
                        out ← r
                w_L ← w_L/√2
            w ← w/√2
```

introduced a variation where a maximum in the Hough accumulator does not prevent its neighbors from generating detections; instead, the votes of the elements of the detected structure are removed from the accumulator before looking for a new maximum. Thus, each element can only belong to a single structure. A similar idea was proposed by Desolneux et al. [2008] under the name of "exclusion principle": The most meaningful observed structure (the one with smallest NFA) is kept as a valid detection. Then, all the basic elements (the points in our case) that were part of that validated group are assigned to it and the remaining candidate structures cannot use them anymore. The NFA of the remaining candidates is re-computed without counting the excluded elements. In that way, redundant structures lose most of their supporting elements and are no longer meaningful. Inversely, a candidate that corresponds to a different structure keeps most or all of its supporting elements and remains meaningful. The most meaningful candidate among the remaining ones is then validated and the process is iterated until there are no more meaningful candidates.

This formulation of the masking process often leads to good results, removing redundant detections while keeping the good ones. But it may also lead to unsatisfactory results as illustrated in Figure 2.8. The problem arises when various valid alignments have many elements in common. As one alignment is evaluated after the other, it may happen that all of its elements have been removed, even if the alignment is in fact not redundant with any of the other ones. In the example of Figure 2.8, individual horizontal and vertical alignments are not redundant, but if all the vertical ones have been detected first, the remaining horizontal ones will be (incorrectly) masked. This example shows a fundamental flaw of the exclusion principle: it is not sound to impose that a basic element belongs to a single perceptually valid structure. This leads us to formulate a relaxed

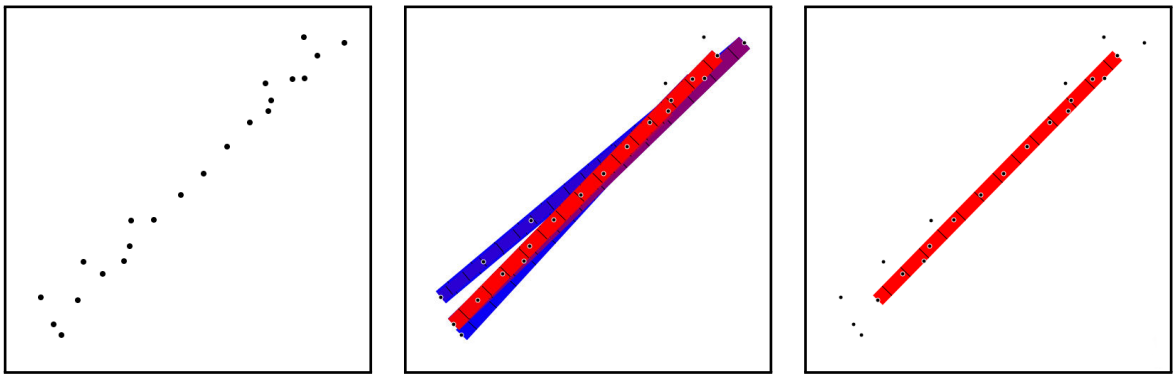


Figure 2.7: Redundant detections. **Left:** point pattern. **Center:** alignments found by Algorithm 4. Red means the most meaningful and blue the least meaningful detections. **Right:** Result of the masking process.

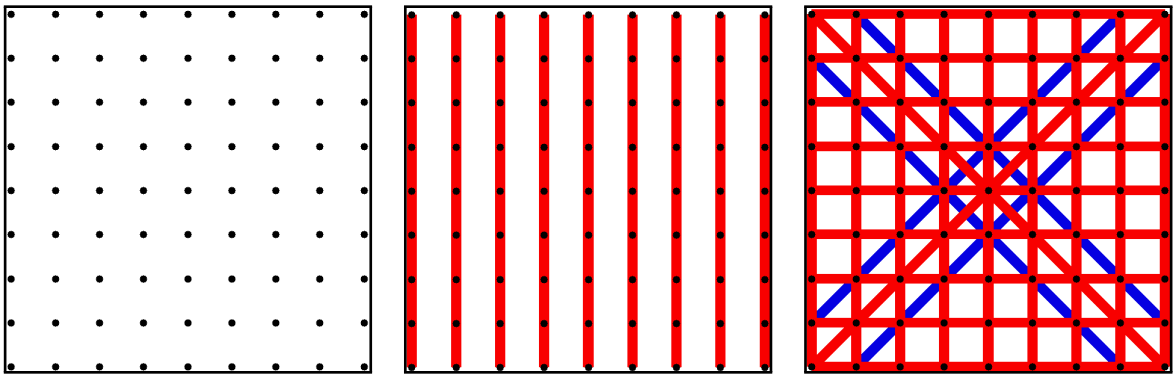


Figure 2.8: Examples of two alternative formulations of the masking process. **Left:** Set of points. **Center:** The Exclusion Principle as defined in Desolneux et al. [2008], a validated gestalt prevents others from using its points. The vertical alignments (evaluated first) mask the horizontal ones. **Right:** The Masking Principle, described in the text, which solves the ambiguities without forbidding basic elements to participate of two different structures.

version of the exclusion principle:

Definition 1 (Building Elements)**.** *We call* building element *any atomic component that can be a constituent element of several structures. An example of building elements are points that can be recursively grouped in alignments.*

Definition 2 (Masking Principle)**.** *A meaningful structure B will be said "masked by a structure A" if B is no longer meaningful when evaluated without counting its building elements belonging to A. In such a situation, the structure B is not retained as detected.*

In short, a meaningful structure will be detected *if* it is not masked by any other detected structure. The difference with the former exclusion principle is that here a structure can only be masked by another *individual* structure and not by the union of several structures. A procedural way to attain this result is to validate alignments one by one, starting by the one with smallest NFA.

Before accepting a new alignment, it is checked that it is not masked by any one of the previously detected alignments.

Figure 2.13 and 2.14 show some point alignment detection results when combining Algorithm 4 with the masking principle. The results obtained in these examples are as expected and this masking procedure was applied to all experiments below. For simplicity, we shall still refer to it as "Algorithm 4".

2.7 Computational Complexity

The aim of this work was to produce an unsupervised algorithm that gives a good solution to the problem, while neglecting at this stage the efficiency considerations. Nevertheless our complexity analysis here gives an upper bound for the existing algorithm. We will also point out several possible accelerations.

The proposed method consists of an exhaustive search for candidates and the validation, including the subsequent redundancy elimination. Algorithm 4 describes the exhaustive search. The number of tests performed is the theoretical number in eq. 3.3. To obtain the total complexity, this number must be multiplied by the complexity of a single test. To compute the NFA we need to evaluate whether each of the N points belongs to a box or not. Thus, the complexity of a single test is proportional to the number of points N. Finally, the total complexity of the exhaustive search is $O(\sqrt{N}N^3)$, where the $\sqrt{N}$ comes from the set of number of boxes tested (C). In the final redundancy reduction step, the validated candidates (n_{val}) need to be compared to those already selected as final detections (n_{out}). This final step has a complexity of $O(n_{val}n_{out}N)$, typically much smaller than the complexity of the exhaustive search.

All in all, the proposed method has a complexity of $O(\sqrt{N}N^3)$. This polynomial time limits the number of points that can be handled in practice. With current computers a few thousands points can be processed.

Several approaches may lead to an effective acceleration of the exploration of candidates, which is the most time consuming step. First, this exploration is highly parallel, leading to an almost linear improvement when using a multiprocessor platform. Our current implementation uses OpenMP, resulting indeed in a near-linear acceleration. Second, there is plenty of room for improvement in the routines for counting the points inside a rectangle; two possible techniques are the use of buckets [Black] or integral images [Crow 1984; Viola and Jones 2001]. Finally, the exhaustive search can be replaced by a smart heuristic such as RANSAC [Fischler and Bolles 1981]. An approach using accelerated versions of the Hough transform [Kiryati et al. 1991, 2000] may be particularly important when the number of points grows. In the next section, we show how the finite mixture algorithm of Figueiredo and Jain [2002] can also be used as an efficient heuristic to propose candidates.

2.8 Acceleration of the algorithm

One way to accelerate the method is to use a fast algorithm to propose alignment candidates, instead of evaluating every pair of points in the dataset. To this end, we propose to use the unsupervised Gaussian Mixtures algorithm of Figueiredo and Jain [2002], exploiting the fact that an elongated cluster can be approximated by a thin Gaussian. This algorithm detects Gaussians of any shape and can automatically discover the number of Gaussian clusters in the dataset (see Figure 2.12 for example results. The clusters found by this algorithm can be considered as potential

alignment candidates and then be evaluated using the NFA.

The computational complexity of Figueiredo's algorithm is similar to the EM algorithm in 2D, which is approximately $\mathcal{O}(nk)$, where k is the number of iterations. Since the algorithm converges in at most a few hundreds of iterations, this is usually much smaller than $\mathcal{O}(n^3)$. To cope with the non-deterministic nature of the algorithm (given by its random initialization), the algorithm can be run multiple times to obtain different lists of candidates.

Algorithm 5 presents the pseudo code for the accelerated version. For each Gaussian detected by Figueiredo's algorithm, we consider a line segment along the longest axis of the Gaussian, centered at the mean and with length twice the square-root of the largest eigenvalue of the co-variance matrix of the Gaussian (lines 6 and 7). The set of endpoints of all such segments are then passed to the alignment detector algorithm, which will compute candidate alignments using only those endpoints (line 9). The `GMM` procedure in line 3 uses the original code of Figueiredo and Jain [2002], obtained in the Author's web page[1].

The `detect_alignments_with_candidate_endpoints` procedure in line 9 of Algorithm 5 implements the alignment detector, except that instead of going through each pair of points to form candidate alignments, it only considers pairs of candidate endpoints obtained with the Gaussian mixtures algorithm.

In Chapter 4, Section 4.3.3 we will use the accelerated version of the algorithm to perform a fast detection of vanishing points in an image.

2.9 Experiments

This section illustrates the proposed algorithm with synthetic data experiments. All the experiments in this section were done with the non-accelerated version of the algorithm. For experiments using the accelerated version please refer to Chapter 4, Section 4.3.3. The reader is invited to perform further experiments using the online demo and source code.[2]

[1] `http://www.lx.it.pt/~mtf/`

[2] `http://dev.ipol.im/~jirafa/ipol_demo/point_alignment_detection/` *(user: demo, password: demo)*

Algorithm 5: Use Figueiredo et al. Gaussian Mixtures algorithm to generate alignment candidates.

input : A list of points $\mathbf{X}$. The number of times the `GMM` algorithm is run, k.
output: A list $\mathbf{A}$ of alignment detections.

1 $\mathbf{X}_c \leftarrow \emptyset$
2 **for** $i = 1 : k$ **do**
3 $\quad (\mu_j, \Sigma_j)_{j=1\ldots n} \leftarrow$ `GMM(`$\mathbf{X}$`)`
4 $\quad$ **for** $j = 1 : n$ **do**
5 $\quad\quad (\mathbf{u}, \sigma, \mathbf{v})_{1,2} \leftarrow$ `SVD(`Σ_j`)`
6 $\quad\quad \mathbf{x}_1 \leftarrow \mu_j + \mathbf{u}_1 \cdot 2 \cdot \sqrt{\sigma_1}$
7 $\quad\quad \mathbf{x}_2 \leftarrow \mu_j - \mathbf{u}_1 \cdot 2 \cdot \sqrt{\sigma_1}$
8 $\quad\quad$ Append $\mathbf{x}_1$ and $\mathbf{x}_2$ to $\mathbf{X}_c$
9 $\mathbf{A} \leftarrow$ `detect_alignments_with_candidate_endpoints(`$\mathbf{X}, \mathbf{X}_c$`)`

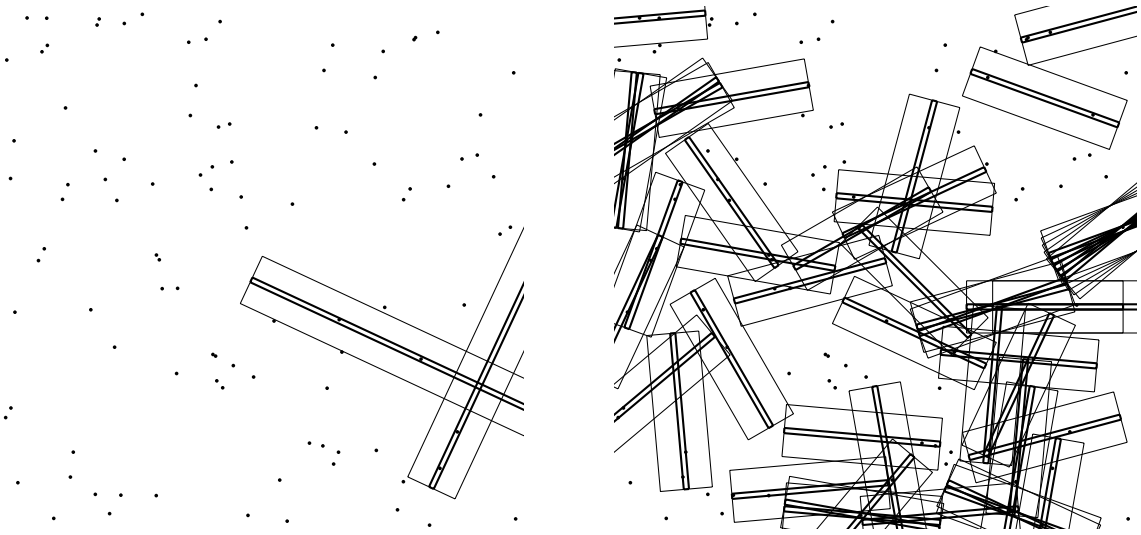


Figure 2.9: Result of Hall et al. [2006] on a set of 100 uniform and independent random points. Each detection is represented by a thin rectangle, surrounded by the local window. **Left:** The same parameters as in [Hall et al. 2006, Section 3.1] were used: 10×10 grid, 5 degree angle step, $a = 0.1$, $b = 0.6$, $c = 0.01$, $u = 6$ $v = 2$. As in Hall et al. [2006], about 3 alignments were detected (2 in this example). **Right:** Result with a slightly different candidate set: 20×20 grid and $b = 0.3$, producing 47 detections. A similar behavior is observed with sets of 1000 random points.

We will first compare the proposed algorithm with two point alignment detection methods [Hall et al. 2006; Hammer 2009]. These two methods were selected because they were introduced recently and include statistical significance tests. Hall et al. [2006] proposed an approach based on similar measurements to Algorithm 3: the alignment is evaluated as a thin strip and two lateral rectangles are used to estimate the point density; the set of candidates and the statistical test are different. The results presented here were computed using our own implementation of the method; we reproduced the experiments in the original publication to verify the correctness of our code.

The statistical test of the method by Hall et al. is designed to reject alignments in a uniform Poisson random point model. The method works well when the intensity of the point process is high. Indeed, the authors showed that the method approaches optimality as the intensity increases [Hall et al. 2006]. The method is less efficient when the density of points is low relative to the size of the operator; in such conditions the sampling is not suitable, the point density estimation is poor, and the statistical test is not able to reject random configurations. The test depends on two parameters, u and v, to be set manually. The first parameter controls the statistical significance level. In extreme cases of wrong density estimation (e.g. no point is observed in the local window), the statistical test fails. The second parameter, v, is a threshold imposed on the number of points in the strip. It is necessary to cope with cases of density under-sampling. The method assumes that the domain is the unit square and tests candidates centered in an $n \times n$ grid, at regular orientations with an angle step θ; three parameters define the strip: the length b, the strip width c, and the local window width a. Thus seven parameters must be provided by the user: n, θ, a, b, c, u, and v.

Figure 2.9 shows two detection results in a set of 100 points generated according to a Poisson model. The first result (left) uses the same parameters as in [Hall et al. 2006, Section 3.1]; in accordance with the results of the original article, in these conditions is observed an average of 3 detections per data set (2 in the example shown). However, when the shape parameters are mod-

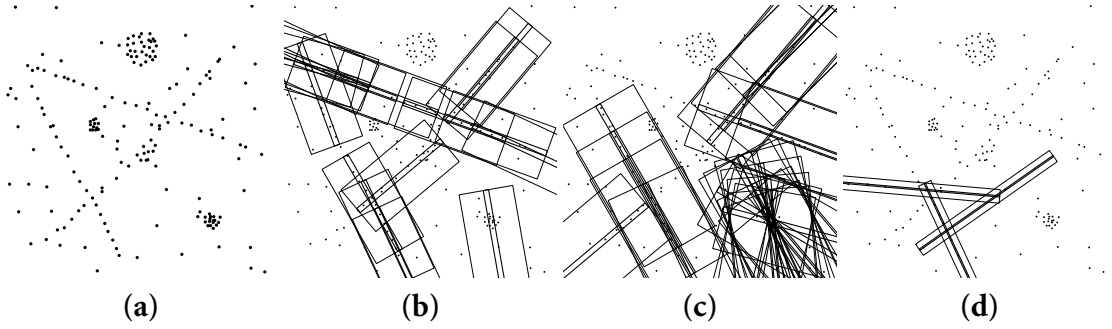


Figure 2.10: Result of Hall et al. [2006]. Each detection is represented by a thin rectangle, surrounded by the local window. In the following results a 20×20 candidate center grid was used, a 5 degree angle step, $u = 7$ and $v = 3$. **(a)** Input set of 186 points. **(b)** Result of the method for $a = 0.2$, $b = 0.4$, and $c = 0.02$. **(c)** Result of the method for $a = 0.3$, $b = 0.6$, and $c = 0.02$. **(d)** Result of the method for $a = 0.05$, $b = 0.6$, and $c = 0.005$.

ified, the statistical test is no longer able to control the number of false detections, see Figure 2.9 (right). In the second experiment, the number of candidates is larger, (the grid is 20×20 instead of 10×10), and the density estimation is worse because the candidates are half as long (thus the local window is half as big). Under the new conditions, the same u and v values lead to 47 detections in the same point set. This experiment shows the need to set manually the statistical significance parameters (u, v) to produce reliable results. This behavior was also observed for sets of 1000 random points.

We will see now how the method handles data sets that do contain point alignments. Figure 2.10(a) shows a set of 186 points; perceptually one can see three alignments, three clusters, and random points. We first adjusted the statistical test parameters u and v so as to obtain very few detections with the same number of random points. The candidate shape parameters (a, b, c) were then adjusted to obtain the best result, see Figure 2.10(b). As one can see, the three alignments were detected. Nevertheless, all detections only partially cover the perceived alignment; this is of course due to the selected length ($b = 0.4$), but longer candidates produced less complete results. Also, there is some redundancy in the detections; no redundancy reduction step is included in Hall et al.'s method. Finally, one can observe that one of the clusters led to a false detection. When the shape parameters are changed to less optimal values, Figs. 2.10(c) and (d), we obtains less useful results: some alignments or parts of them are missing, and many spurious detections were produced. Some are due indeed to deviations from the random model as is the case of the clusters, and this shows the need for a more complex event definition. Others, as in Figure 2.10(d), reveal a failure of the statistical test. This experiment shows that the shape parameters need a careful adjustment to produce good results.

The second point alignment detection method we used for comparison was introduced by Hammer [2009]. This method requires an alignment length parameter (defined as a radius). Each point of the input set defines a candidate. The distribution of angles from the center point to each of the points inside the radius is evaluated. When the circular-uniform distribution is rejected, using a Rayleigh test, the candidate produces a detection. The last decision requires a significance level α.

The experiments presented here were done using the author implementation of the algorithm, included in the software package PAST [Hammer et al. 2001]. Figure 2.11 shows the results for the three data sets considered in this comparison and for four parameter sets (no detection was pro-

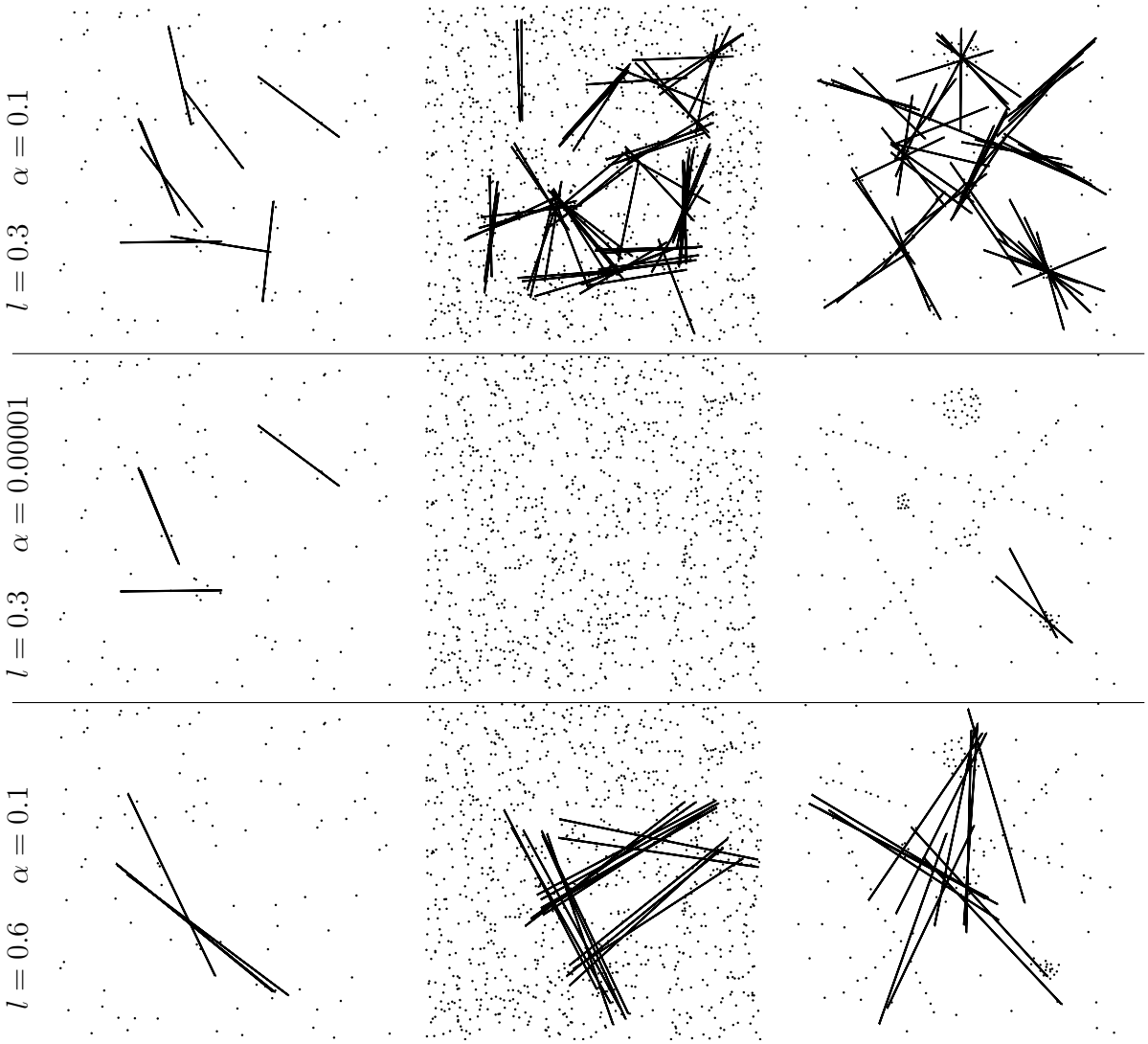


Figure 2.11: Result of Hammer Hammer [2009] for three point sets in a normalized unit square domain. Two length values were tested: 0.3 (radius 0.15) and 0.6 (r. 0.3); and two significance levels: $\alpha = 0.1$ and $\alpha = 0.00001$. No detection was produced with $l = 0.6$ and $\alpha = 0.00001$. **Left:** The same set of 100 random points used in Figure 2.9. **Middle:** 1000 points drawn independently with uniform distribution in a unit square. **Right:** The same point set as in Figure 2.10.

duced for $l = 0.6$ and $\alpha = 0.00001$). As one can see in the first row, the default significance level used in PAST is not satisfactory: it produces many detections on random sets of points (left and middle). Using this significance level one gets some of the expected alignments in the data set containing alignments (right). But, as in the previous method, redundancy is observed and many false detections, mainly caused by the presence of the clusters. When the significance level is changed to $\alpha = 0.00001$ (second row), the number of false detections in noise is reduced significantly; unfortunately, the true alignments also disappear, leaving only two detections due to a cluster. Using a longer alignment length (third row) produced no better results. Increasing the significance level for long alignments led to no detection.

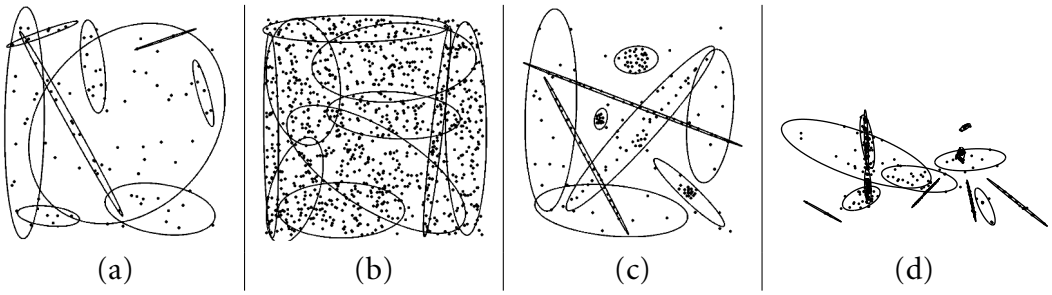


Figure 2.12: Unsupervised clustering by the Figueiredo and Jain [2002] method. Ellipses represent detected clusters. **(a), (b), (c)** The sets of points in Figures 2.9, 2.10, 2.11. **(d)** Dataset from a vanishing point detection problem, see Chapter 4.

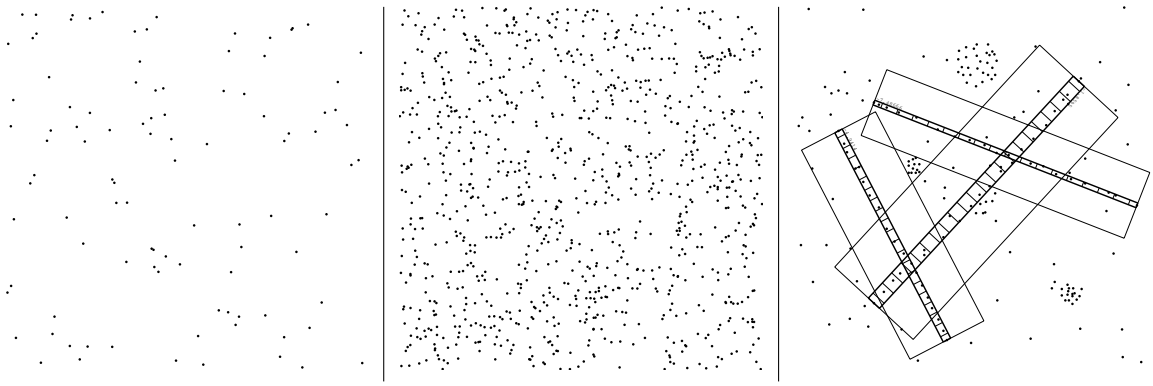


Figure 2.13: Result of the proposed algorithm for the sets of points in Figures 2.9, 2.10, 2.11 and 2.12. Each detection is represented by a thin rectangle divided into boxes, and surrounded by the local window.

As discussed in Section 2.1, general clustering methods can provide point alignments when a criterion is added to select elongated clusters. We will show results of this approach using the well-known algorithm by Figueiredo and Jain [2002]. This algorithm adds an important aspect to our comparison: like our algorithm it is unsupervised. Figure 2.12 shows the results for the same point sets used before. Being a randomized algorithm, the best results obtained in our tests are presented. The method was used in its standard form, fitting Gaussian mixtures. Each ellipse in the figure corresponds to a Gaussian cluster. The results obtained for the structured points are surprisingly good: each one of the perceived alignments and clusters is well represented. The middle result on random points is far less satisfactory as it includes many elongated clusters interpretable as alignment detections.

Figure 2.13 shows the results of the proposed algorithm for the same data sets. As one can see, no detection is produced in the random points, and the three alignments were found. Two of the detections are however shorter than expected and the top vertex of the "A" is missing. Note how the method correctly handled the redundant detections. Some further results of our method, with increasing difficulty, are shown in Figure 2.14, where the figures are correctly solved. Notice how the very low relative density alignment in Figure 2.14 (right) was correctly detected.

To conclude this experimental section, we present and comment some example figures that show the limitations of our algorithm, see Figure 2.15. Again, we will use synthetic datasets that

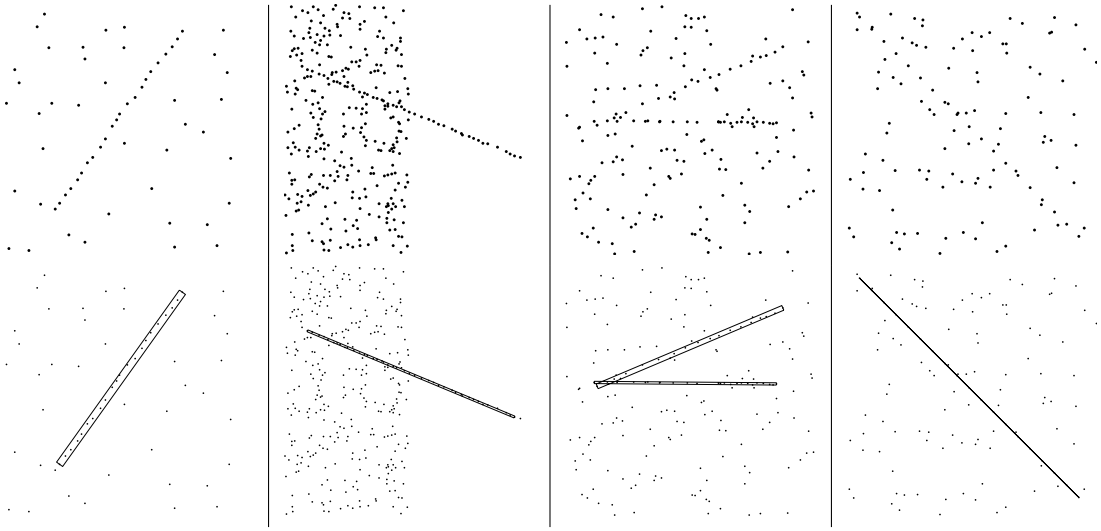


Figure 2.14: Additional results of the proposed algorithm. The local window and the boxes are not drawn.

show clearly each condition, but similar effects can be observed in real data. All the alignments in Figure 2.15(a) were found; however, the redundancy reduction step did not select the candidates covering the complete alignments. Also, a global interpretation of the figure is lacking but requires other detection tools. The set of points in Figure 2.15(b) is the same already shown in Figure 2.1; the alignment found by the algorithm is correct, but as discussed before, does not correspond to the most common interpretation by a human observer. A natural way of handling this problem would be to detect the "curves" by good continuation and then forcing a global interpretation by meth-

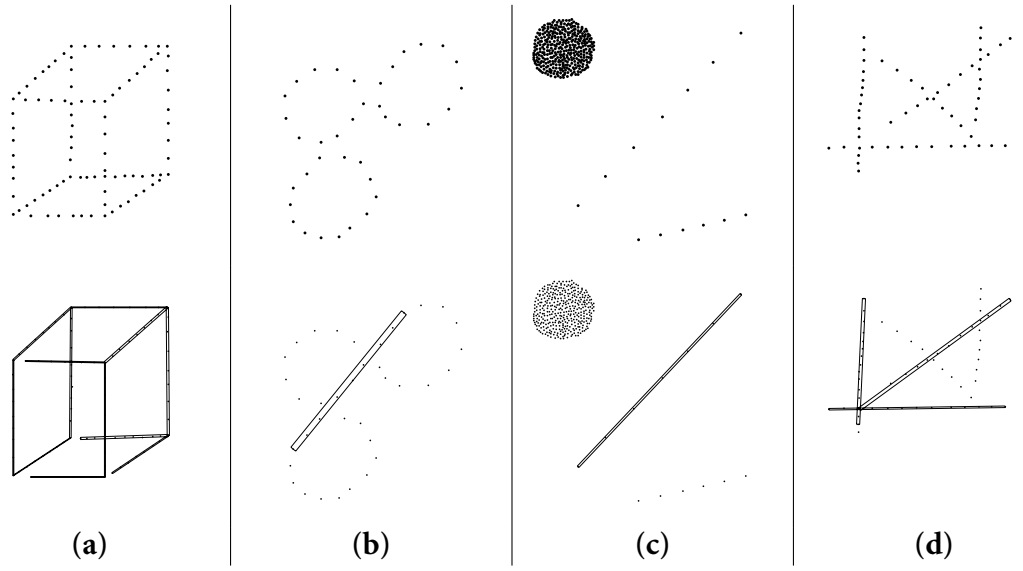


Figure 2.15: Examples imperfectly solved by the proposed algorithm. The local window and the boxes are not drawn.

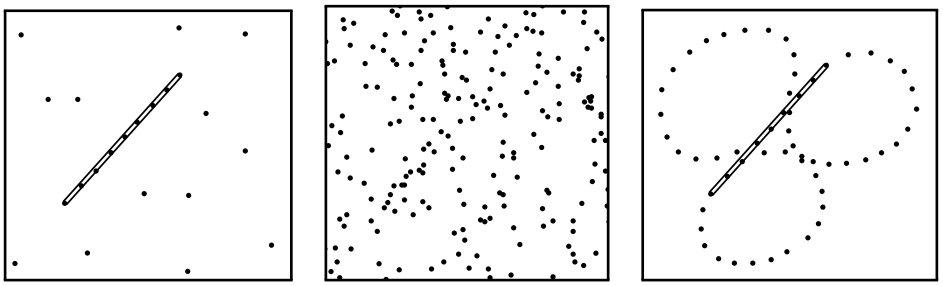


Figure 2.16: Results of the alignment detector in the examples of Figure 2.1.

ods similar to our masking methodology, that would discard the detected alignment as "masked" by the curves. In Figure 2.15(c) the presence of a large cluster masks an alignment: it increases artificially the number of tests and short segments are no longer detected. Handling this example probably implies a round cluster detector but also a recursive approach: once a cluster is detected and removed, our algorithm would easily detect both alignments. Finally, two of the structures in Figure 2.15(d) are slightly curved, rendering them inaccurate as alignments. Again, the detection of good continuation may provide a solution. In short, no alignment detection algorithm can be fully satisfactory *per se*; it requires the interaction (and conflicts) with other feature detectors.

2.10 Conclusion

In this chapter we have presented a series of algorithms with incrementing sophistication for detecting alignments of points in a point pattern. The two key aspects of the alignment have been shown to be its local density and its regularity. Our final method combines both criteria into a single coherent detector. We have also introduced a new procedure to resolve the problem of redundant detections. Finally, we have presented successful results on both synthetic and real datasets. The main limitation of the method is its high computational cost, for which we propose an improvement using an auxiliary algorithm to compute alignment candidates. In Chapter 4 we will apply the method to a classic problem, the detection of vanishing points. When a *gestalt conflict* occurs, as in Figure 2.1 (right), the method fails to give the correct interpretation. In an attempt to solve this problem, the same methodology will be used to handle the detection of the *Good Continuation* gestalt in Chapter 3.

3 On Good Continuation Detection

In this chapter we propose a model for a more general visual task: the grouping of dots by good continuation. Naturally, this model is inspired by the findings of Chapter 2, taking into consideration points regularity and local density. Our model for good continuation is based on local symmetries, and as before, the non-accidentalness principle to determine perceptually relevant configurations. A robust, scale-invariant and unsupervised algorithm for the detection of good continuation of points is derived. The method is supported by experimentation on traditional Gestalt figures, as well as in dot pattern datasets existing in the literature. The application of our method to the detection of dotted lines on scanned documents is also illustrated. Finally, we compare our method to the successful Tensor Voting approach in figure-ground segmentation tasks. We note three drawbacks of Tensor Voting that are corrected in our method: scale-invariance, robustness to noise, and a principled heuristic to extract final curves.

The detection algorithm produced in this chapter can be tested online on any dot pattern at
`http://dev.ipol.im/~jlezama/ipol_demo/lgrm_good_continuation_matlab/`
(user: demo, password: demo)

3.1 Introduction

The Gestalt school of psychology [Wertheimer 1923; Metzger 1975; Kanizsa 1979; Wagemans et al. 2012a,b] assumes the existence of a short list of grouping laws governing visual perception. Among them, the law of *good continuation* can be stated as "All else being equal, elements that can be seen as smooth continuations of each other tend to be grouped together" [Palmer 1999]. Figure 3.1 exemplifies this law; a perceptual organization of this image would result in a three part configuration: a line, an arc of circle, and a zigzag, all formed by dots. Unfortunately, this law, as the other Gestalt laws, was enunciated only qualitatively, without a formalization into a predictive framework.

Since it was first enunciated by Wertheimer [1923], the Gestalt law of good continuation has been one of the most extensively studied perceptual phenomena. Various aspects of this law have been examined, including amodal completion and contour integration of basic oriented and unoriented elements. In this work we concentrate in the case of unoriented elements.

The advantage of working with unoriented elements (such as dots), is that the grouping and masking processes are isolated from the appearance of the basic elements. Experiments using dot patterns have been explored by many works in psychophysics. Notably by the early works of French

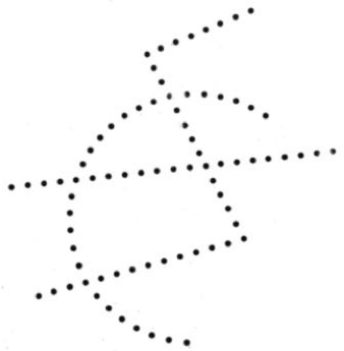


Figure 3.1: *Good Continuation* law: human perception tends to group elements on a smooth, continuous order. Image extracted from Kanizsa [1979].

[1954], Uttal et al. [1970], and Uttal [1973] but also in more recent works [Mussap and Levi 2000]. Uttal studied the influence of length, dot spacing, curvature and outlier noise in the perceptual grouping of dot structures. Regularity in dot patterns was analyzed by Feldman [1997a]; Kubovy et al. [1998]. In the famous work of Glass et al. [1969], superposed random dot patterns were used to show the importance of local interactions for the building of global percepts.

The particular case of perceptual grouping by good continuation in dot patterns is also covered in a vast literature. Prinzmetal and Banks [1977] use dot patterns to prove the existence of the good continuation phenomenon. Early algorithmic proposals for modeling the phenomenon [Caelli et al. 1978; van Oeffelen and Vos 1983; Smits et al. 1984; Smits and Vos 1986] consisted in convolving the dot patterns with a Gaussian kernel to find groupings by thresholding the result of this convolution. This idea, formalized in the CODE algorithm of van Oeffelen and Vos [1983], had a relative success and was further extended in psychophysics [Compton and Logan 1993; Logan 1996] as well as in computer vision, as will be discussed below. More recent approaches analyze the curvature of the curves generated by successive dots. In Feldman [1997b] the probabilistic properties of successive angles in a perceived chain of dots is studied. Pizlo et al. [1997] proposed a clever pyramidal system to account for local-global interactivity.

Another approach to the study of the good continuation law is through the following simple experiment: dots are arranged along a virtual circular contour that produces a linear interpolation when the number of points is small and a curvilinear interpolation when the number is large [van Assen and Vos 1999]. The exact number of dots needed to pass from one type of interpolation to the other is studied with psychophysical experiments. In a similar line, Gori and Spillmann [2010] take a collinear arrangement of dots, and modify the spacing between given pairs, studying the boundary between the perception of an irregular alignment and its splitting into multiple segments.

The law of good continuation applied to oriented elements has also been extensively studied in psychophysics, notably in Field et al. [1993], who use Gabor patterns [Machilsen and Wagemans 2011; Demeyer and Machilsen 2012] or line segments [Feldman 2007] as the oriented elements. Strongly related to the good continuation gestalt is the closure gestalt. Many works by Elder and Zucker state the importance of contour closure for the perception of structures, see for example Elder and Zucker [1993, 1998]. A quantitative measure of closure and an algorithmic approach are proposed in Elder and Zucker [1994, 1996].

In the computational domain, many algorithms inspired by the good continuation law have been developed. Early proposals tried to define a global vision mechanism detecting multiple

gestalts [Grossberg and Mingolla 1987; Carpenter and Grossberg 1987]. Sha'asua and Ullman [1988] applied in a remarkable early algorithm the good continuation law to identify salient image features. A saliency map is obtained by iterative local computations on the image edge elements that minimize an energy favouring smooth and long curves. Parent and Zucker [1989] proposed a rigorous and clever approach to inferring curves as a labeling problem, based on local interactions that favor co-circularity (which is a form of local symmetry, a notion that our method also exploits). The image elements are convolved with oriented filters formed with Gaussians, to determine tentative tangent orientations. In Gigus and Malik [1991], the convolution with oriented Gaussians is also exploited, simply taking the maximum filter responses. This has the algorithmic advantage of being non-iterative, although detection performance may be affected. In Herault and Horaud [1993] the problem of figure-ground segmentation is posed from a combinatorial optimization perspective, and it is solved with simulated annealing. Another groundbreaking work is the Tensor Voting approach introduced by Guy and Medioni [1992, 1993], which proposes a saliency measure that involves the summation of vector votes emitted by each element. The votes encourage co-circularity and proximity of the elements. Unlike the other approaches, this framework specifically incorporates the detection of curves of unoriented elements, so we will expressly compare our method to theirs in Section 3.5. This approach was continued in various works [Mordohai and Medioni 2006; Loss et al. 2006, 2009; Gong and Medioni 2012]. Perona and Freeman [1998] pose the figure/ground segmentation as a factorization of a matrix representing the affinity between elements. Williams and Thornber [1999] provide a very good review of existing approaches and introduce a new one, where the saliency measure is given by the number of times a random walk passes by an edge, and where the transition probability matrix is also given by the affinity matrix. Dubuc and Zucker [2001a,b] presented a principled framework for curve characterization based on differential geometry. In general, all these methods are based on finding combinations of local interactions that favor curve smoothness, length and elements proximity.

In this work, we propose a new model and algorithm for the grouping by good continuation (restricted to unoriented elements), using a simple model that favors local symmetries, and with a detection control based on the non-accidentalness principle. This allows the method to be general in the sense that it can capture curves of any shape and scale, and is robust to the presence of outliers. It is also unsupervised because detections are given by their statistical significance, which requires only a single parameter, namely the number of false detections that would be allowed in an image of random noise.

The proposed algorithm consists of two main steps: building candidate chains of points, and validating them. Candidate chains of points are built by considering triplets of points formed by joining nearest neighbors. Once valid triplets have been obtained, a graph representation is produced where each node corresponds to a triplet. A classical path finding algorithm is run on this graph to obtain paths between all pairs of triplets. Finally, the paths found are validated or rejected using thresholds obtained with the *a contrario* approach [Desolneux et al. 2008].

This chapter is organized as follows: Section 3.2 presents our proposed mathematical model of good continuation chains. Section 3.3 describes an efficient algorithm for detecting good continuation configurations in dot patterns. The mathematical model and the algorithm are then evaluated in Section 3.4 and compared to the Tensor Voting approach in Section 3.5. Section 3.6 presents the conclusions of this study.

3.2 Mathematical Model

Let us consider a set of N planar points. The aim is to find a mathematical model that can predict when an ordered subset of points lie on a smooth curve that is salient relative to the background of the other points, see Figure 3.2(a). Each ordered subset of points (a sequence of points) will be called a *chain*; each set of three consecutive points in a chain will be called a *triplet*. The proposed model is based on the idea that the better the symmetry of the triplets, the better the saliency of the sequence. Ideally, the third point in a triplet should be symmetric to the first, relative to the middle point, in the position marked with an X on Figure 3.2(b). The better the symmetry of the triplets, the better the smoothness of the chain. Symmetric triplets also enforce a second Gestalt grouping law: *proximity* [Wagemans et al. 2012a]. A regular spacing of the points along the curve favors their perceptual grouping; inversely, an irregular spacing [Wertheimer 1923; Gori and Spillmann 2010] would tend to stop the curve at larger gaps.

The precision of a triplet will be measured by the distance between the observed third point and its ideal symmetric position X, see Figure 3.2 (c). A whole chain will be characterized by the number of triplets in it and their worst precision. The considered event is then: a chain of k triplets, each one with precision r_{max} or better.

To obtain a perceptually plausible model, the evaluation of a chain will be again based on the *a contrario* methodology, as was done in Chapter 2. This *a contrario* methodology has already been applied at detecting good continuations of image edges [Cao 2004]. Even if related, the main source of information for this detector was the orientation of the edges, while no orientation is associated to points in our case. The method of Cao was successfully extended to encode shapes for performing *a contrario* shape matching and recognition [Musé et al. 2006; Cao et al. 2007].

We will evaluate the probability of observing a triplet of a given precision under a random hypothesis. To enforce scale-invariance, this probability will be evaluated relatively to the context contained in a circular local window L with radius R, where R is proportional to the triplet size, see Figure 3.2(c). Given that n points were observed in L (not counting the first two of the triplet,

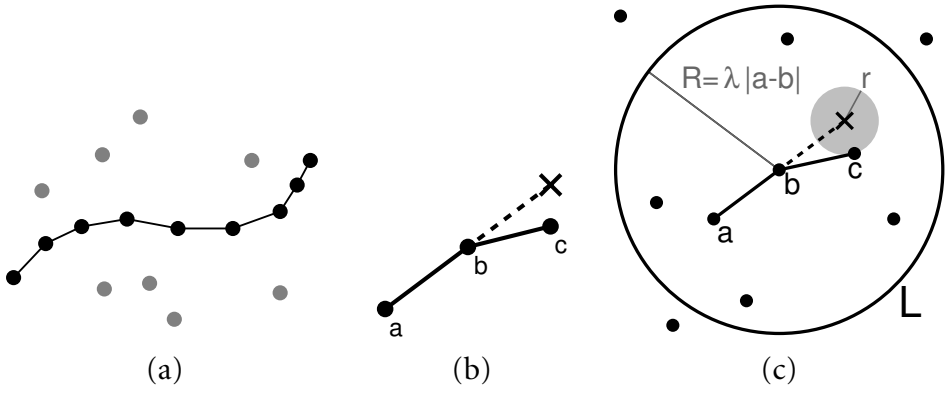


Figure 3.2: Definition of the *good continuation* event. **(a)** A candidate *chain* is defined by an ordered sequence of points. **(b)** Three consecutive points in a chain define a *triplet*, (a, b, c) in this case. Ideally, the triplet should be symmetric. That is, the third point c should be symmetric to the first a relative to the middle point b. The ideal third point is represented by X. **(c)** The symmetry precision of a triplet is measured by the distance r from the ideal point X to its nearest point. When evaluated relative to a local window L of radius R, this precision can also be expressed as the probability that, among the n points in the local window, the nearest point to X be at most at a distance r.

because they define the local window), our random or *a contrario* model H_0, used to evaluate accidentalness, is that these points are independent and uniformly distributed in L. In other words, our *a contrario* model assumes that the n points result from a spatial uniform Poisson process in L. This *a contrario* model H_0 is not intended to model the statistics of the sought structure; quite the opposite, it models random data where the sought structure is not present, and is used to calibrate rejection thresholds.

Under these assumptions, we would like to translate the precision of each triplet into probabilistic terms. Let us call ρ the distance between the ideal point X and its nearest point in L. We will evaluate the precision of a triplet by the probability $\mathbb{P}(\rho \leq r)$, for the observed radius r. It is simpler to compute its complement, $\mathbb{P}(\rho > r)$, which implies that all the n points in L fall outside the disk of radius r; given that L is a disk of radius R, $\mathbb{P}(\rho > r) = \left(1 - \frac{\pi r^2}{\pi R^2}\right)^n$. Finally, a triplet is associated a probability

$$p = \mathbb{P}(\rho \leq r) = 1 - \left(1 - \frac{r^2}{R^2}\right)^n. \tag{3.1}$$

On what follows we shall interchangeably use the words precision and probability of a triplet to refer to p.

Consider a chain $\mathcal{C}$ of k points $a_1, a_2, \ldots, a_k$. The precision p_i of each of the $k-2$ triplets (a_i, a_{i+1}, a_{i+2}) will be evaluated, and the worst case value, $p_{\max} = \max\{p_1, p_2, \ldots, p_{k-2}\}$, is associated to the whole chain. Recall that the event we are considering is a chain $\mathcal{C}$ of $k-2$ triplets, each with precision $p_{\max}$ or better. To compute the probability of this event we will use the fact that, under the *a contrario* Poisson assumption, the precision of each triplet is independent from the previous ones. Finally, the probability of observing the $k-2$ triplet with precision $p_{\max}$ is:

$$\mathbb{P}(\mathcal{C}) = p_{\max}^{k-2}. \tag{3.2}$$

The NFA for a chain of points in *good continuation* is computed as:

$$\text{NFA}(\mathcal{C}) = N_{tests} \cdot \mathbb{P}(\mathcal{C}). \tag{3.3}$$

We kindly remind the reader that the NFA is an upper bound to the expected number of events as good as $\mathcal{C}$ to be observed by chance in the *a contrario* model H_0. A large NFA means that such an event was to be expected under the *a contrario* model and therefore is irrelevant. On the other hand, small NFAs correspond to rare events and therefore arguably meaningful.

The number of tests N_{tests} is the number of chains considered as potential good continuations. The proposed method will generate candidates starting at each of the N points and using each of the b nearest neighbors as a second point. Using these two points, the ideal point X is constructed (see Figure 3.2(b)) and the closest point to it is selected as the third point. Using the last two points we construct the new symmetric point and we repeat this process until a maximal chain length of $\sqrt{N}$ is reached. Here we assume that a smooth 1D subset of a 2D set of N points would be typically limited to $\sqrt{N}$ points. Each of the intermediate chains is evaluated and counted as a test. Thus, the number of tests is $bN\sqrt{N}$. Finally, the NFA of the event "having k points in *good continuation* configuration up to a precision $p_{\max}$" is defined by:

$$\text{NFA}_5 = bN\sqrt{N} \cdot p_{\max}^{k-2}. \tag{3.4}$$

Let us call $\mathcal{C}_{i,j}^K$ the chain that starts at point i, continues at the j^{th} nearest neighbor of i, and has K points. Note that this chain, defined by i, j and K, is unique because each successive point

in a chain is the closest point to the ideal symmetric point defined by the last two points. Let us consider $\mathbf{X}$, a random set of points under H_0. We will refer to $\text{NFA}_5\left(\mathcal{C}_{i,j}^K, \mathbf{X}\right)$, as the expected number of occurrences in $\mathbf{X}$ of an event with the characteristics of the observed chain $\mathcal{C}_{i,j}^K$. These characteristics are determined by the worst precision triplet in the chain and they consist of the local window size R, the number of points in it n, and the distance between the ideal point and the closest point to it, r.

Theorem 5.

$$\mathbb{E}\left[\sum_{i=1}^{N}\sum_{j=1}^{b}\sum_{K=3}^{\lceil\sqrt{N}\rceil+2}\mathbb{1}_{\text{NFA}_5\left(\mathcal{C}_{i,j}^K,\mathbf{X}\right)\leq\varepsilon}\right]\leq\varepsilon$$

where $\mathbb{E}$ *is the expectation operator and* $\mathbb{1}$ *is the indicator.*

Proof. We define $\hat{r}(R,n,K)$ as

$$\hat{r}(R,n,K)=\max\left\{r\in\mathbb{N},\mathbb{P}\left[\rho(\mathbf{X})\leq r|R,n\right]\leq\left(\frac{\varepsilon}{bN\sqrt{N}}\right)^{\frac{1}{K-2}}\right\}. \tag{3.5}$$

Here, R is the radius of the local window and n is the number of points in it. The radius ρ is the distance between the ideal symmetric position and its closest point. The probabilistic model inside the local window, conditioned to the fact that the number of observed points is n is still uniform and independent, and the conditional law of the number of points inside any subset of R follows a binomial law. Then, $\text{NFA}_5\left(\mathcal{C}_{i,j}^K,\mathbf{X}\right)\leq\varepsilon$ is equivalent to $\rho(\mathbf{X})\leq\hat{r}(R,n,K)$ for all triplets in the chain. Now,

$$\mathbb{E}\left[\sum_{i=1}^{N}\sum_{j=1}^{b}\sum_{K=3}^{\lceil\sqrt{N}\rceil+2}\mathbb{1}_{\text{NFA}_5\left(\mathcal{C}_{i,j}^K,\mathbf{X}\right)\leq\varepsilon}\right]=\sum_{i=1}^{N}\sum_{j=1}^{b}\sum_{K=3}^{\lceil\sqrt{N}\rceil+2}\mathbb{P}\left[\text{NFA}_5\left(\mathcal{C}_{i,j}^K,\mathbf{X}\right)\leq\varepsilon\right]=$$

$$\sum_{i=1}^{N}\sum_{j=1}^{b}\sum_{K=3}^{\lceil\sqrt{N}\rceil+2}\sum_{n_1=1}^{N}\ldots\sum_{n_{K-2}=1}^{N}\int_{R_1=0}^{\inf}\ldots\int_{R_{K-2}=0}^{\inf}\mathbb{P}[\rho_1(\mathbf{X})\leq\hat{r}(R_1,n_1,K),\ldots$$

$$\ldots,\rho_{K-2}(\mathbf{X})\leq\hat{r}(R_{K-2},n_{K-2},K)|R_1,n_1,\ldots,R_{K-2},n_{K-2}]\cdot\mathbb{P}(R_1,n_1,\ldots,R_{K-2},n_{K-2})=$$

$$\sum_{i=1}^{N}\sum_{j=1}^{b}\sum_{K=3}^{\lceil\sqrt{N}\rceil+2}\sum_{n_1=1}^{N}\ldots\sum_{n_{k-2}=1}^{N}\int_{R_1=0}^{\inf}\ldots\int_{R_{K-2}=0}^{\inf}\mathbb{P}\left[\rho_1(\mathbf{X})\leq\hat{r}(R_1,n_1,K)|R_1,n_1\right]\cdot\ldots$$

$$\ldots\cdot\mathbb{P}[\rho_{K-2}(\mathbf{X})\leq\hat{r}(R_{K-2},n_{K-2},K)|R_{K-2},n_{K-2}]\cdot\mathbb{P}(R_1,n_1,\ldots,R_{K-2},n_{K-2}). \tag{3.6}$$

In the last step we have used the independence of the triplets. By definition of $\hat{r}(R,n,K)$

$$\mathbb{P}\left[\rho(\mathbf{X})\leq\hat{r}|R,n\right]\leq\left(\frac{\varepsilon}{bN\sqrt{N}}\right)^{\frac{1}{K-2}}.$$

So

$$\prod_{t=1}^{K-2}\mathbb{P}\left[\rho_t(\mathbf{X})\leq\hat{r}(R_t,n_t,K)|n_t,R_t\right]\leq\frac{\varepsilon}{bN\sqrt{N}}.$$

Noting that $\sum_{n_1=1}^{N} ... \sum_{n_{K-2}=1}^{N} \int_{R_1=0} ... \int_{R_{K-2}=0}^{\inf} \mathbb{P}(R_1, n_1, ..., R_{K-2}, n_{K-2}) = 1$:

$$\mathbb{E}\left[\sum_{i=1}^{N}\sum_{j=1}^{b}\sum_{K=3}^{\lceil\sqrt{N}\rceil+2} \mathbb{1}_{\mathrm{NFA}_5\left(\mathcal{C}_{i,j}^{K},\mathbf{X}\right)\leq\varepsilon}\right] \leq \sum_{i=1}^{N}\sum_{j=1}^{b}\sum_{K=3}^{\lceil\sqrt{N}\rceil+2} \frac{\varepsilon}{bN\sqrt{N}} = \varepsilon$$

which concludes the proof. □

Given an observed set of N points and a candidate chain of k points, we will consider the latter event as an ε-meaningful *good continuation* when the corresponding NFA is lower than ε. It can be shown [Desolneux et al. 2008] that the expected number of events with NFA $\leq \varepsilon$ is bounded by ε in the *a contrario* model H_0. As in Chapter 2, we will always fix $\varepsilon = 1$ to ensure an unsupervised detection.

3.3 Algorithm

This section describes an efficient algorithm for searching meaningful chains of dots using the model presented in the previous section. Given an input of N 2D points, the candidate chains are obtained by exploring the b nearest neighbors of each point to construct candidate triplets and then by connecting paths between every two triplets. To find the paths, a graph representation of the triplets is constructed and the Floyd-Warshall algorithm is used. Each path found is a candidate chain that is finally evaluated using the NFA measure (3.4), and the most significant chains are kept using a redundancy reduction step, described below. This process is described in Algorithm 6. Note that the candidate search is not performed exactly as the theoretical exploration described in the previous section to obtain the number of tests, so the value in (3.4) is an approximation. However, this approximation proves to work well in practice.

The algorithm requires two parameters: the number of nearest neighbors b, used for exploration and λ, the proportion of the local window size to a triplet's size[1] (see Section 3.2). Lines 1 to 11 of Algorithm 6 build the list $\mathcal{T}$ of triplets to be considered. Note that each triplet is stored with its precision.

A pair of triplets will be called *adjacent* when they share two points in such a way that they can form a chain of four points. (Triplets that share two points but form a "Y" shape are not adjacent.) We define a graph where triplets are the vertices and adjacent triplets s and r share an edge with value $p_s + p_r$, the sum of their precisions. Lines 12 to 19 compute the distance matrix $\mathcal{D}$ that defines this graph.

Once the adjacencies are determined, the algorithm uses the Floyd-Warshall algorithm to compute the path with shortest distances between every two vertices. (Notice that this "distance" is not Euclidean, but is a sum of triplet precisions; thus the classic bias toward small objects is not present here.) This will provide the best chain joining every pair of triplets, in the sense of the smallest sum $\sum_i p_i$ along the path. This yields a redundant set of candidate paths, among which to select the paths with lowest NFA. Note that there is no theoretical guarantee that paths with minimal NFA are all contained in a minimal path for the Floyd-Warshall algorithm. However experimental results show that this approximation is acceptable. A similar procedure was proposed in Elder and Zucker [1996] for the application of detecting closed contours, where a graph representation of edge elements is used, and shortest paths are searched for to compute closed contours.

[1] All the results shown in this chapter use $b = 5$ and $\lambda = 4$.

Algorithm 6: *Good Continuation* detection

Input: A list of N 2D points, b the number of NN used for exploration and λ the proportion of the local window radius w.r.t. the triplet size.

Output: A list of *good continuation* chains $\mathcal{GC}$

1 $\mathcal{T} \leftarrow [\,]$
2 **for** *point* $i = 1$ *to* N **do**
3 **for** *point* $j \in \mathrm{NearestNeighbors}_b(i)$ **do**
4 Count n, nbr. of points in the local window L centered in j with radius $R = \lambda \cdot dist(i, j)$
5 Compute s_j^i, the symmetric of i w.r.t. j
6 **for** *point* $k \in \mathrm{NearestNeighbors}_b(j)$ **do**
7 Compute $P_{i,j,k} = \mathbb{P}(\rho < dist(s_j^i, k) | R, n)$, using (3.1)
8 $\mathcal{T} \leftarrow \big(i, j, k;\ P_{i,j,k}\big)$
9 $\mathcal{D} \leftarrow$ a $|\mathcal{T}| \times |\mathcal{T}|$ matrix of ∞
10 **for** *triplet* $r \in \mathcal{T}$ **do**
11 **for** *triplet* $s \in \mathcal{T}$ **do**
12 **if** r *is adjacent to* s **then**
13 $\mathcal{D}_{r,s} \leftarrow \mathcal{T}_r^p + \mathcal{T}_s^p$
14 $\mathcal{P} \leftarrow$ Floyd-Warshall$(\mathcal{D})$
15 **for** *path* $p \in \mathcal{P}$ **do**
16 **if** $\mathrm{NFA}(p) \leq \varepsilon$ **then**
17 $\mathcal{GC} \leftarrow p$

The computational complexity of the Floyd-Warshall algorithm is $O(|V|^3)$, where $|V|$ is the number of vertices in the graph, i.e., the number of triplets. In terms of computation time, this is the bottleneck of the proposed algorithm. The result is a non-linear algorithm, but fast enough in practice. Finally, all the candidate chains provided by Floyd-Warshall will be evaluated for significance using (3.4) and the ones with NFA $\leq \varepsilon$ will be kept (lines 21 to 25).

Once all the *good continuation* events are found, we are interested in keeping only the maximal meaningful events. Note that one *good continuation* event might mask another smaller event contained in itself. For simplicity, we say that an event A masks an event B, if $\mathrm{NFA}_A < \mathrm{NFA}_B$ and the chains share at least two points. The latter is a simple criterion to allow crossing chains, which share one point. This trivial solution has been found empirically. Obtaining a list of only the most meaningful events, which are not masked by any other event, can be done by following the simple steps described next. First, the meaningful chains are ordered by their NFA (lowest first). A second list is created which in the beginning contains only the most meaningful chain. Then, the first list is traversed, checking if each chain is masked with any of the chains in the second list. If a chain is not masked, it is added to the second list.

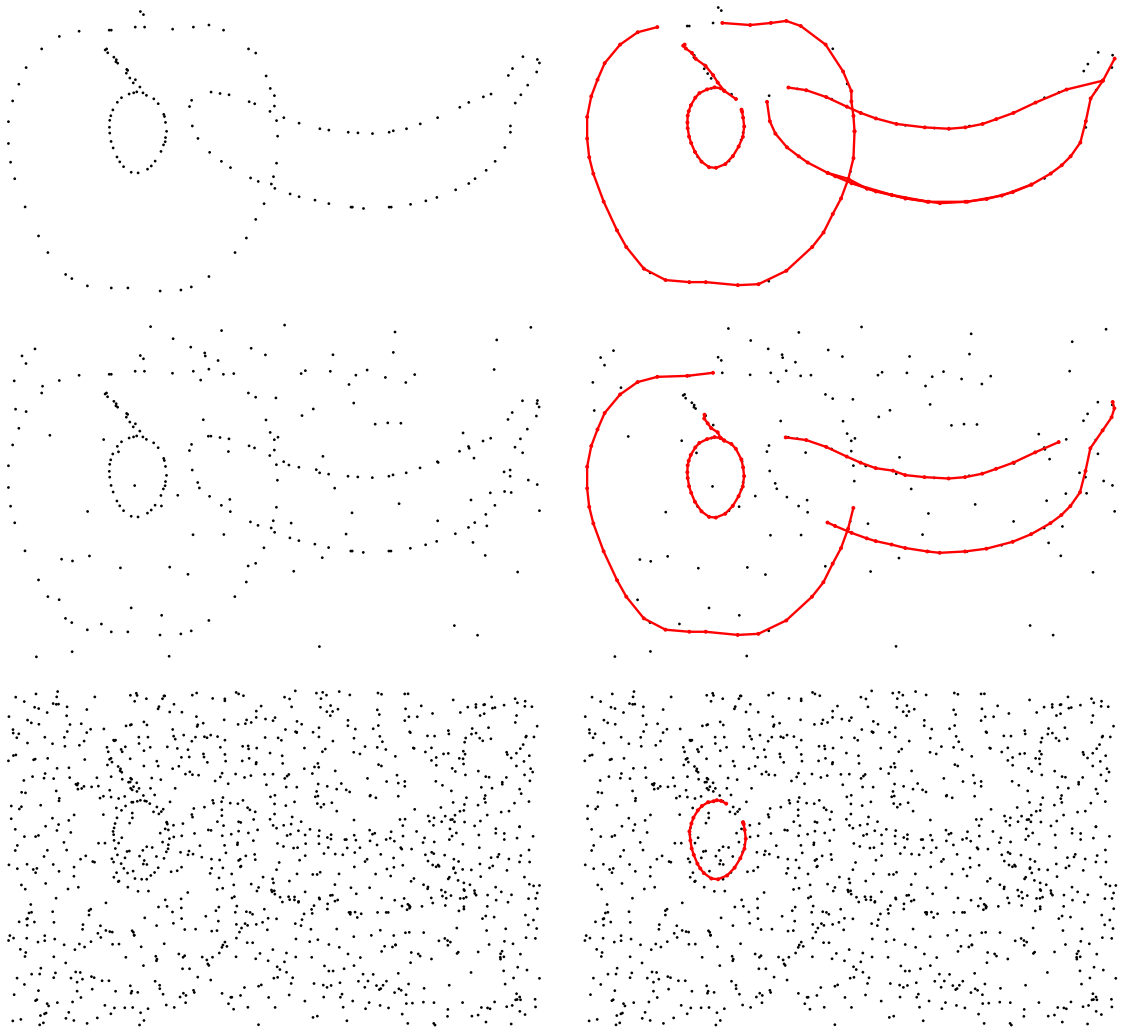


Figure 3.3: Result of the algorithm using dot patterns adapted from the "yellow apple", "banana" and "tamarillo" edge patterns from Williams and Thornber [1999]. The first row shows three figures from the original dataset, where the orientation information has been dropped and the figures have been rescaled and displaced. In the second and third row 100 and 1000 random points are added, respectively. The resulting detections are consistent with the interpretation a human observer may do of the figure.

3.4 Experiments

Figure 3.3 presents examples of detection results obtained with the algorithm described in Section 3.3. We took three images from the fruit and vegetables dataset of Williams and Thornber [1999]: and kept only the position of the oriented elements, discarding the orientation information. The examples are taken from silhouettes of an apple, banana and a tamarillo. To demonstrate the scale invariance of our approach, we scaled each image, so they are at 1/3 and 1/6 scales respectively. Next, we added 100 random points to the figure, to test the robustness of our approach in differentiating perceptually relevant structure from noise. Finally, we added 1000 random points, which visually mask the two largest figures. In this case, only the contour of the tamarillo is still

detected by the algorithm, because of its higher dot density. One may argue that the borders of the banana are still slightly perceived, but in this case the observer might be influenced by the previous knowledge of the target or by other gestalts such as closure.

In the rest of this section we will perform a detailed analysis of the NFA obtained with our model through multiple examples, showing its applicability as a quantitative perceptual measure. Next, we will present the algorithm result for dot pattern figures taken from the good continuation literature. In the next section we will compare the algorithm to the Tensor Voting approach.

The reader is invited to try the online demo of this algorithm to further extend this experimental section[2].

3.4.1 Analysis of the NFA

To analyze the effect of curvature, dot density and dot regularity in the NFA of a chain, we considered the dot pattern examples of Uttal [1973] (Figure 1). We analyzed the NFA obtained for each curve using equation (3.4). To demonstrate the perceptual plausibility of the NFA, we added two types of noise: a surrounding outlier noise and inlier noise implemented as a random jitter on the position of the points.

Figure 3.4 shows the original figures scanned from Uttal [1973] in the left column. To save space, we have removed the fourth line of the experiments from the original article. To recover the position of the dots we used a Harris corner detector [Harris and Stephens 1988] on the scanned images. On the right, we show the NFA obtained for each of the curves. Note how the NFA decreases as the curvature increases or as the dot density of the curves decreases. If we set the meaningfulness threshold to $\varepsilon = 1$ (one false alarm in average in noise) curves #5 and #6 of experiment 3, are no longer meaningful. This is because the angle in the middle, which will determine the precision of the event, is too closed. In this case, our model would prefer to split the curve into two straight segments. This can be seen in the actual result of the algorithm in Figure 3.7, which shows the most significant curves. Still, for a human observer, it is possible that a higher-level grouping process produces the junction of both segments.

Figure 3.5 shows the same curves of Figure 3.4 with 20 random points added to each one. The aim of this experiment is to show how the NFA correctly models the masking/unmasking perception process. The NFA of the curves is less meaningful in the presence of noise, but it is still meaningful where the structure can still be perceived. On the other hand, when the noise is sufficient to mask the structure, (notably in experiments 2 and 3 where the structure has fewer points), the structure becomes statistically as well as perceptually indistinguishable from noise, and the NFA goes above 1. This observation is in line with the original conclusions of Uttal [1973], although the experiment setup is by no means the same.

The experiments of Figure 3.6 aim at showing the effect in the NFA of the irregular placement of dots. Starting with the original curves of Figure 3.4, we added isotropic random displacements to each dot. The random displacements are Gaussian distributed, centered at each dot and with a 4 pixels standard deviation (as a reference, inter-dot distance is approx. 20 pixels). The results show a degradation of the NFA with respect to the original curves. This was to be expected because the local symmetry is strongly violated. In particular, the results of the experiment 1, curve #6 and experiment 2, curves #5 and #6 have an NFA above the meaningfulness threshold of 1. When looking at those curves, a human observer might find it easier to interpret them as two separate pieces of curves instead of a single one.

[2] `http://dev.ipol.im/~jlezama/ipol_demo/lgrm_good_continuation_matlab/`

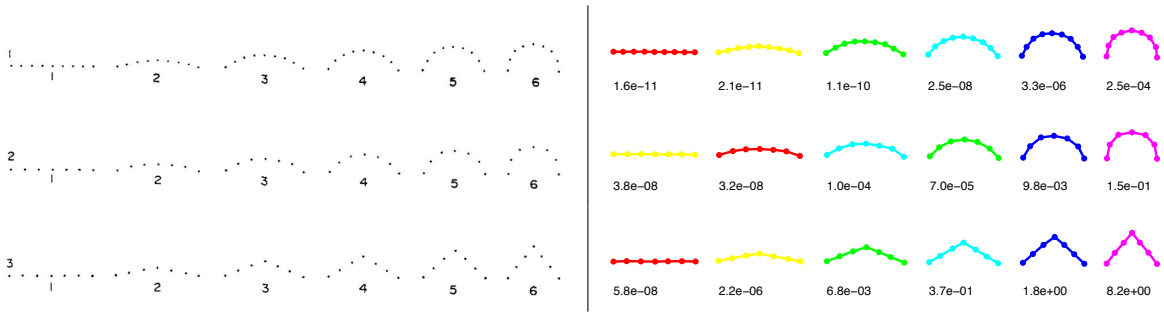


Figure 3.4: NFAs for the curves used in Uttal [1973]. The points have been obtained by scanning the figure and running a Harris corner detector [Harris and Stephens 1988]. On the left column, the NFA obtained for each curve is shown.

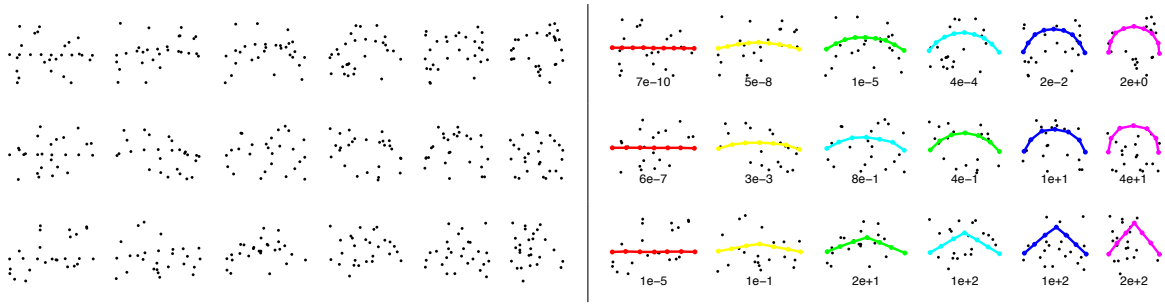


Figure 3.5: NFAs for the curves of Figure 3.4 plus 20 outliers. Note how the NFA decreases in the presence of noise, sometimes above the meaningfulness threshold. The NFA as a quantitative predictive measure is consistent with the perception of the curves in the left figures.

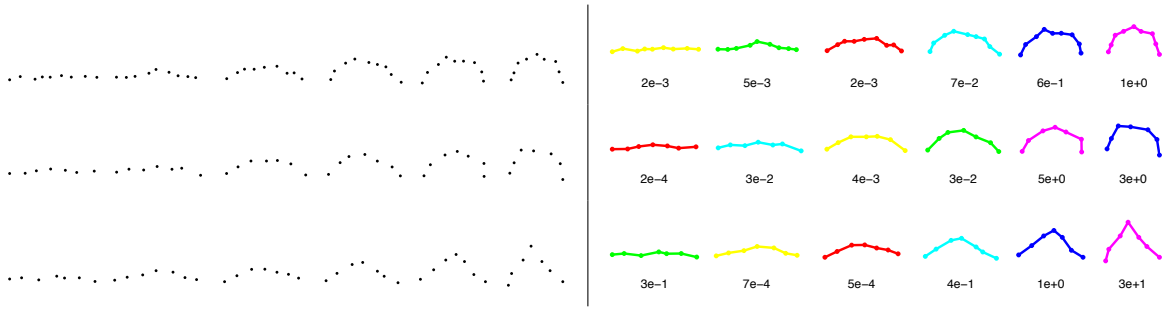


Figure 3.6: NFAs for the curves of Figure 3.4 with Gaussian jitter ($\sigma = 4px$) on the points position. The NFA is degraded because the local symmetries are deteriorated. In some cases a human observer might prefer to split the curve in two. In general, for those cases the NFA is above the meaningfulness threshold.

3.4.2 Results of the algorithm

Figure 3.7 shows the result of the good continuation detection algorithm described in section 3.3 for the dot patterns of Figures 3.4, 3.5 and 3.6. The figure is divided in three groups according to the type of noise: no noise, outlier noise, and inlier noise.

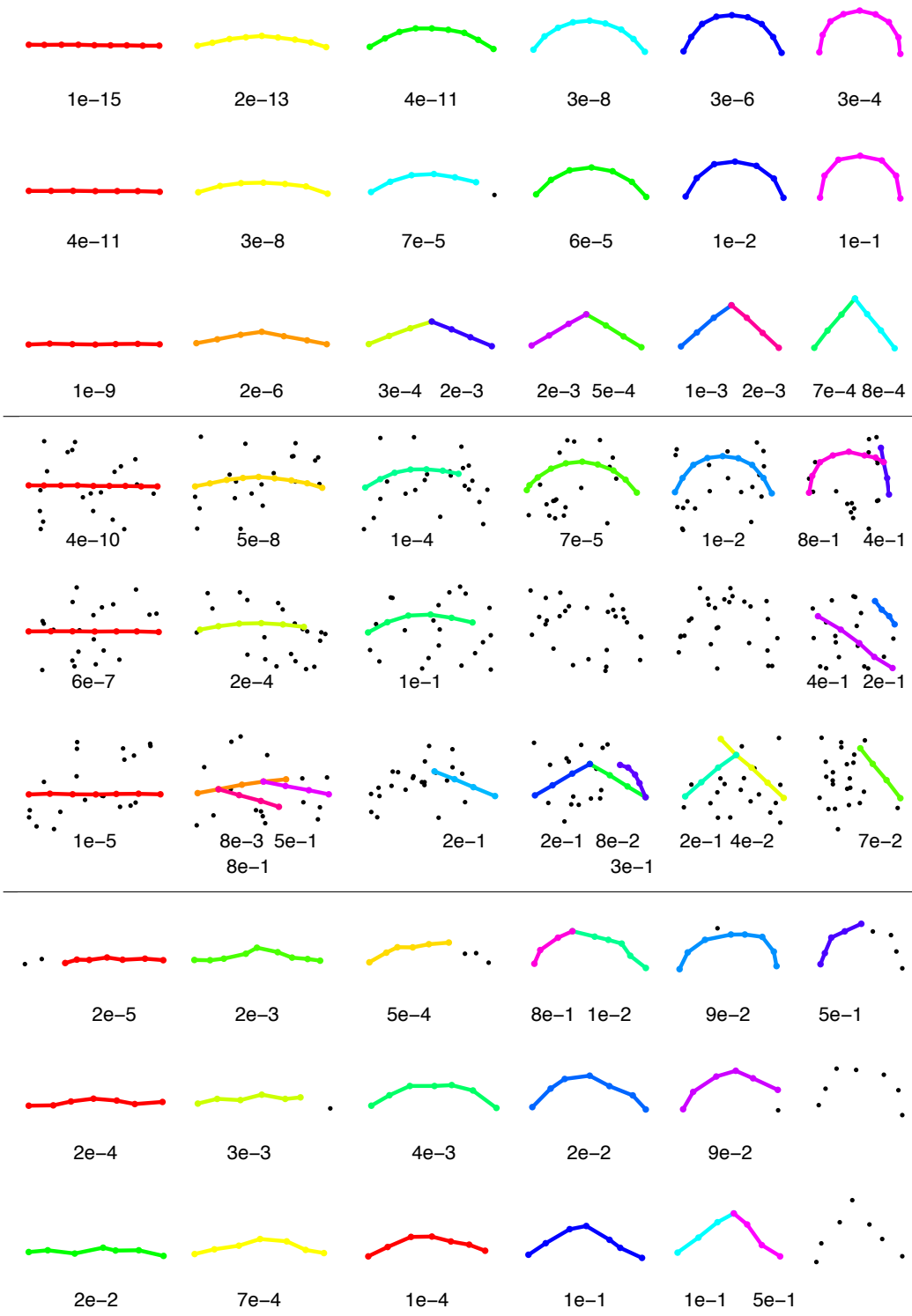


Figure 3.7: Result of the algorithm for the images of Figures 3.4, 3.5 and 3.6. The experiments are grouped by the type of noise: none, outlier and inlier noise.

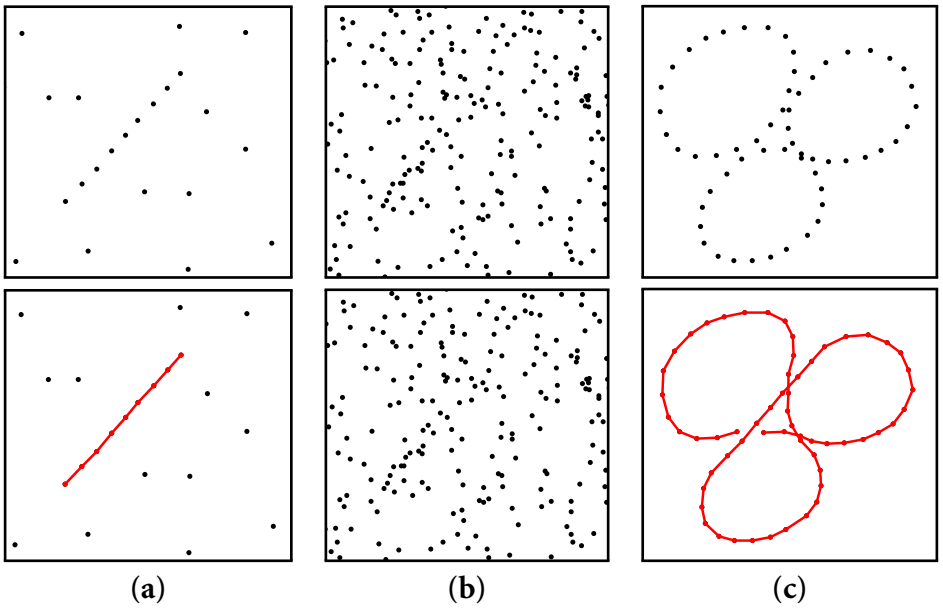


Figure 3.8: Results of the good continuation detector in the examples of Figure 2.1. The first and second examples are correctly solved by the algorithm. In the third example, the result is successful in terms of finding the best good continuation curve. However, most viewers describe this figure as a set of three convex closed curves. The incorporation of other gestalts such as closure and convexity seems to be required.

In the noise-free experiments, the algorithm finds the original curve in most cases. One exception is experiment 2, curve #3, where the algorithm prefers the curve that is formed by leaving out the last dot. Note that not all triplets along the curve are exactly equal because of position noise due to the scanning and Harris detector. What is happening is that the precision of the last triplet is such that the NFA would increase rather than decrease if it were included, so the most significant curve leaves that dot out. Indeed, the detection algorithm only keeps the most significant curve among all candidates. In experiment 3, when the angle is too strong, the separate segments to each side are more significant than the entire curve. In the second group of experiments, where outlier noise is present, the algorithm tends to find the curve where it is still perceived. Otherwise, there are two reasons for a curve not to be detected. The first trivial reason is when the NFA is simply too large and therefore not meaningful, because the triplet's probabilities increase as more points are present in the local window (see Figure 3.2(c)). The second reason is that due to the algorithm's heuristic of searching among the nearest neighbors, the curve may never be considered as a candidate, and never be evaluated. This is why for some curves the NFA as it would be calculated by an ideal observer is meaningful, but they are not detected by the algorithm. Examples of this case can be found in experiment 2, curves #3 and #4. This effect is actually perceptually plausible: when there are many points in the image, the complexity of evaluating every possible combination is arguably intractable for human perception.

In the third group of experiments, where random jitter is added to the points positions, the algorithm produces splits where a triplet precision is too coarse. In those cases, separate segments of the curve are individually more meaningful than the whole curve. One can observe that those splits are also perceptually plausible. Another interesting case is experiment 1, curve #5, where the algorithm prefers to leave one point out of the curve. Again, looking at the input dots this

interpretation seems natural.

Figure 3.8 shows the result of the good continuation detector for the examples of Figure 2.1. The result for the first and second examples is perceptually plausible. The result for the third example is correct in the sense that all points participate in a meaningful curve. However, this global interpretation of the figure is different from what a human observer would probably perceive, which is three distinct closed curves. The incorporation of the closure and convexity gestalts, seems to be required. Modeling and implementing these gestalts remains as future work.

Figure 3.9 shows further results of the algorithm in dot patterns obtained from scanning figures from popular articles in the literature. The input points to the algorithm were once again the points obtained by running a Harris corner detector. Note that spurious corner detections are also part of the input, which causes reasonable detections where text is present. In (a), the algorithm would rather split the inner spiral in two. Although this is a possible explanation, human perception tends to see the complete curve. In this case, incorporating convexity could improve the result. Finally, in Figure 3.10 we show a possible computational application of the method: the automatic processing of graphs from scanned documents.

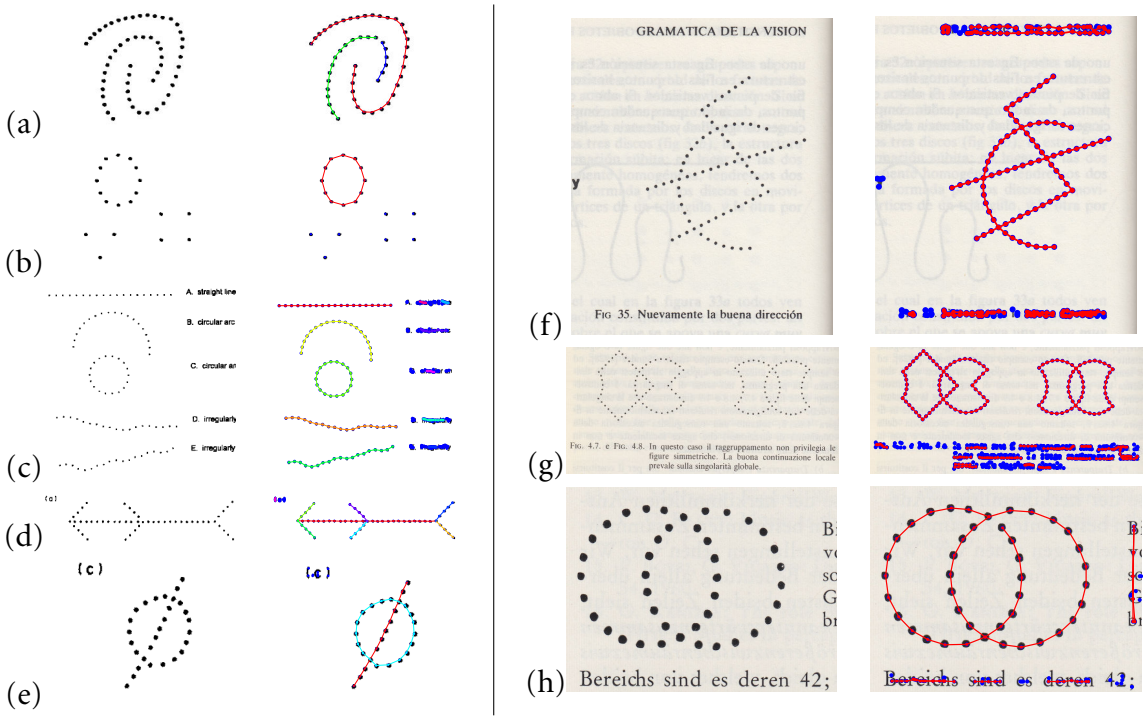


Figure 3.9: Result of the good continuation detection algorithm for images scanned from the following articles: (a) & (b) van Oeffelen and Vos [1983]. (c) Pizlo et al. [1997]. (d) & (e) Caelli et al. [1978]. (f) Kanizsa [1991]. (g) Kanizsa [1980]. (h) Metzger [1975]. On each scanned image in the left column, a Harris corner detector was run to find the dots. The right column shows the Harris detections in blue, and the good continuation configuration found by our method in red.

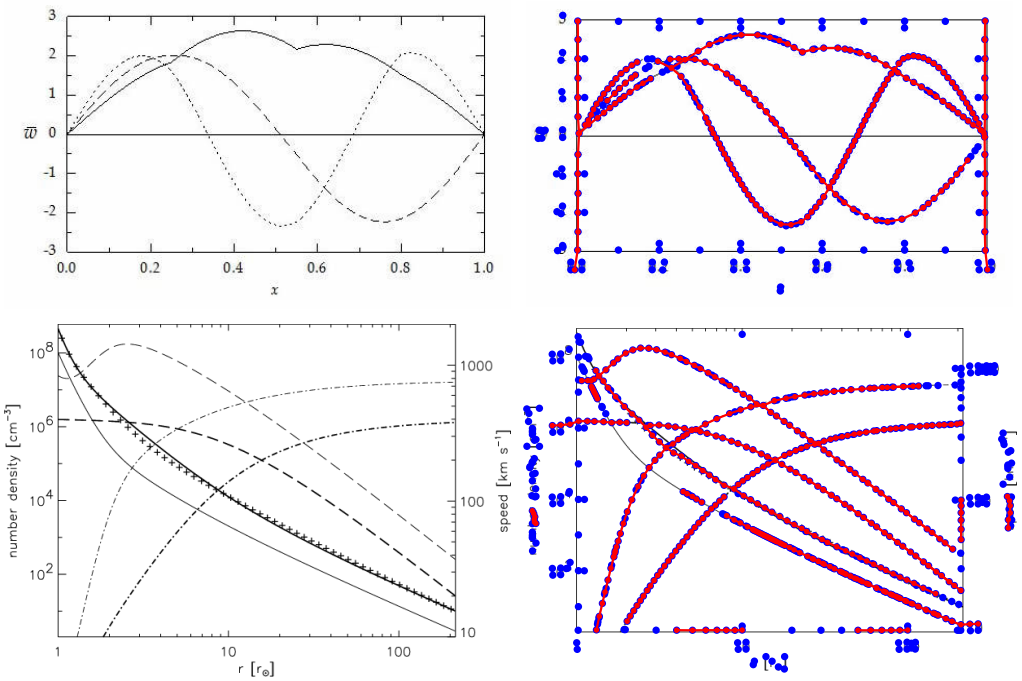


Figure 3.10: A possible application for the method is the processing of scanned documents. In the left column we show graphs scanned from real documents. In the right column, Harris corner detections are represented by blue dots and curve detections in red.

3.5 Comparison to Tensor Voting

In this section we compare our method to a well established method in the literature for finding salient configurations of points, the Tensor Voting approach.

There exists a varied literature of algorithmic approaches to good continuation perceptual grouping, as discussed in Section 3.1. However, all of the aforecited methods work with oriented elements, which provide a fundamental cue on the direction of the *good continuation*. Since the method presented in this chapter concentrates on the perceptual grouping of unoriented elements, a comparison with methods using oriented elements would require a non-trivial adaptation of them to make the comparison fair. Such adaptation would also require further supervision.

Robust, unsupervised clustering of points is a long standing problem [Jain 2010]. In the family of non-parametric clustering methods, spectral clustering is a successful technique [Ng et al. 2002; Von Luxburg 2007] that could in some cases resolve the grouping by good continuation (see for an example Figure 1 of Ng et al. [2002]). However, the method requires the number of clusters to be known in advance, and is not able to distinguish structure from noise since it is based on local interactions. Most clustering methods require the number of clusters to be known (e.g. k-means) or a scale parameter (e.g. mean-shift [Comaniciu and Meer 2002]). Unsupervised methods that try to automatically discover the number of clusters also exist, [Figueiredo and Jain 2002; Dy and Brodley 2004]. The work of Figueiredo and Jain [2002] in particular is very successful in finding the number of clusters, and it could assign background noisy samples to a big spread Gaussian. However, as most methods of this kind, it is only applied on a given parametric distribution of the points (for example Gaussian mixtures) so we also excluded it from our comparison.

The detection of non-parametric curves formed by points can also be seen from a manifold learning perspective: curves are 1D manifolds embedded in a 2D space. Most multiple manifold learning techniques can only handle the assumption that the manifolds are linear or affine subspaces [Elhamifar and Vidal 2009; Yan and Pollefeys 2006]. The multiple, non-linear case is treated in Souvenir and Pless [2005]; Goh and Vidal [2007]; Wang et al. [2011], but without the presence of outliers. In Gong et al. [2012], the problem of multiple manifold learning with outliers is elegantly addressed. However, contrarily to our method, the number of manifolds (curves in this case) is required.

Tensor Voting (TV) [Guy and Medioni 1992; Mordohai and Medioni 2006] is a successful algorithmic approach for perceptual grouping that provides a principled way of working with unoriented elements. The TV algorithm can be applied to point patterns, yielding a saliency map where configurations of points in good continuation are given a high intensity. Moreover, for each point in the map, it provides the direction of the tangent to the salient curve. Recently, the TV approach has been extended to account for inlier noise in the Probabilistic Tensor Voting work of Gong and Medioni [2012]. Since our dataset does not include inlier noise, we only compared to standard TV. In the following we shall briefly describe the TV approach. We kindly refer the reader to Guy and Medioni [1992]; Mordohai and Medioni [2006] for a gentle introduction to the method.

In TV, the 2D input elements are represented as structure tensors or 2x2 matrices. The largest axis of the tensor is aligned with the tangent of the curve structure to which an element belongs. The input elements communicate with each other in order to derive the most preferred orientation information (or refine the initial orientation if given) for each of them. The communication is done in the form of a voting field cast by each point, whose magnitude decays with distance and curvature according to a scale parameter σ. Then, vote accumulation is performed by tensor (or matrix) addition, and the saliency map is obtained as the difference between the tensor eigenvalues $\lambda_1 - \lambda_2$. When the ellipsoid represented by the tensor is elongated ($\lambda_1 \gg \lambda_2$), this means that the voting produced a favorable orientation in the major direction of the tensor. When the ellipsoid approaches a disk ($\lambda_1 \simeq \lambda_2$) or when both eigenvectors are small, the orientation at that point is not well defined and thus not considered part of a salient structure. Note that a non-maximal suppression step is required to extract the salient structures.

In this section we shall only compare to the TV algorithm, since it is the most successful method proposing a formal solution to the perceptual grouping of unoriented elements. We obtained one implementation of this algorithm online[3]. To obtain the most salient curves, we performed hysteresis thresholding in the saliency map, along the direction of the tangent vectors. The obtained curves that pass through less than three points were discarded. We selected thresholds that gave the best results in all datasets in average, for each scale.

In our analysis, we found that the TV method has three main drawbacks: The need to set the scale parameter, the harmful effect of surrounding noise and the lack of a principled method to extract the final curves in a figure. The method proposed in this work deals with these drawbacks. First, scale invariance is obtained by using a local window proportional in size to the point triplet. Second, robustness to noise is guaranteed by setting the detection threshold *a contrario*. Third, the extracted final curves have a statistical meaning, formalizing the "non-accidentalness" principle.

3.5.1 Datasets

In order to quantitatively evaluate our method, we used point patterns from three publicly available point pattern datasets. We also extended two of them by transforming the data and adding

[3] `http://www.mathworks.com/matlabcentral/fileexchange/21051-tensor-voting-framework`

Name	No. of figs.	Description
FV1	9	original points from Williams and Thornber [1999]
FV2	9	FV1 + 75% outlier noise
FV3	9	FV3 + 150% outlier noise
FV4	9	three different figures from FV1, rescaled and translated
FV5	9	FV4 + 50% outlier noise
FV6	9	FV4 + 100% outlier noise
HS1	200	original points from Al-Maadeed et al. [2012]
HS2	200	HS1 + 50% outlier noise
HS3	200	HS1 + 100% outlier noise
CR1	100	"Chinese character" points from Chui and Rangarajan [2003]
CR2	100	"fish" points from Chui and Rangarajan [2003]
RN1	10	250 random points
RN2	10	500 random points
RN3	10	1000 random points

Table 3.1: Description of the dot pattern datasets used to quantitatively compare our good continuation detector with Tensor Voting.

noise, and we considered a fourth dataset, containing only random points. The performance of the grouping algorithms was evaluated by their ability to distinguish noise from structure.

The first dataset was formed using the edge images of the fruit and vegetable edges dataset of Williams and Thornber [1999]. It includes 9 different edge figures of fruits and vegetables obtained from real photos. This dataset has been previously used in perceptual grouping benchmarks [Williams and Thornber 1999; Loss et al. 2006, 2009]. Since this work concentrates on grouping unoriented elements, the edge directions were discarded and only the location of the edge elements were kept. We have extended this dataset in two ways: First by adding 75% and 150% outlier noise (by this we mean 75% and 150% more points, randomly distributed). Second, by taking three different figures from the original set, changing their scale so that the first figure is at scale 1, the second at scale 2/3 and the third at scale 1/3 and displacing them to different locations in a new image. Finally, we created two more datasets by adding 50% and 100% outlier noise to the sets of multiple figures. Example figures from this dataset are shown in the top two rows of Figure 3.11 and in the top row of Figure 3.12.

The second dataset was created from the handwritten signature data of Al-Maadeed et al. [2012]. This dataset has been proposed for a signature trajectory prediction challenge in Kaggle[4]. It consists of sets of 2D points marking the trajectory of handwritten signatures captured by a drawing tablet. Although the signature trajectory prediction is a possible extension of our algorithm, this work concentrates only on recovering perceptually salient structures. Thus, we just interpret the signatures as perceptually salient curves. From the 605 images available in the Kaggle challenge, we selected 200 images that were not heavily affected by quantization. We extended this dataset by adding 50% and 100% outlier noise. Example figures from this dataset are shown in the third and fourth rows of Figure 3.11.

[4] `http://www.kaggle.com/c/icdar2013-stroke-recovery-from-offline-data`

The third dataset uses the point patterns introduced in the context of point matching with outliers by Chui and Rangarajan [2003]. In Chui and Rangarajan [2003], two point patterns are used: "Chinese character" and "fish". In the original work, the point patterns are corrupted with deformations and outlier noise. We took the part of the dataset that contains deformations and 50% outliers. This makes in total 100 figures for the "Chinese character" and 100 figures for the "fish" pattern. One example figure for the fish pattern can be seen in the bottom row of Figure 3.11.

Finally, we produced a fourth dataset that contains only random points, consisting of 30 images with 250, 500 and 1000 random points (10 images for each number of points). The intent of this dataset is to show how our detector correctly determines that there is no structure in noise.

In each dataset, we labeled each point as "structure" or "noise". The error was calculated as the proportion of misclassified points. Table 3.1 summarizes the datasets used for quantitative evaluation. Figures. 3.11 and 3.12 show example figures.

3.5.2 Results

Table 3.2 shows the average error results obtained for our method and TV with three different scale parameters. The error measure is the rate of misclassified points. For TV, we tried multiple scales, and multiple threshold values. We have selected the threshold values that worked best for all datasets in average. Note that the performance of TV is sensitive to the scale parameter, which means that a given value for the scale parameter performs better or worse depending on the dataset. For example, without the presence of outliers, choosing a scale too big and a saliency threshold too low is not a problem, since all points are salient. In the presence of noise however, a scale too big and a threshold too low would create false saliency detections, whilst a small value for the scale and

Dataset	TV $\sigma = 0.04$	TV $\sigma = 0.08$	TV $\sigma = 0.12$	Ours
FV1	23.8%	15.6%	11.7%	**3.0 %**
FV2	29.0%	20.4%	15.2%	**8.8 %**
FV3	31.5%	26.4%	22.8%	**13.8%**
FV4	13.6%	13.2%	19.5%	**7.6 %**
FV5	19.8%	20.1%	24.7%	**14.0%**
FV6	27.3%	24.5%	25.2%	**16.2%**
HS1	26.9%	33.2%	44.7%	**14.1%**
HS2	30.2%	32.9%	38.8%	**17.5%**
HS3	34.2%	33.1%	35.2%	**18.9%**
CR1	38.6%	41.1%	49.1%	**21.0%**
CR2	23.6%	26.1%	33.1%	**10.4%**
RN1	40.0%	29.2%	16.8%	**0.5%**
RN2	42.7%	31.4%	19.3%	**0.6%**
RN3	40.1%	28.7%	17.2%	**0.0%**

Table 3.2: Average rate of misclassified points over all the figures in each dataset. For TV at each scale, we have selected the thresholding values of the saliency map that give the best average result across datasets.

a high value for the saliency threshold could miss relevant structures.

Our algorithm does not require parameter tuning and was run unsupervised for all datasets. The scale invariance comes from the fact that the local window for density estimation is proportional to the spacing of the dots along the curve. The extraction of final curves is done by thresholding the NFA of the curves at $\varepsilon = 1$, which means that in average, only one false detection is obtained in noise. Our approach attempts to formalize the "non-accidentalness" principle: meaningful curves in terms of the NFA tend to be perceptually relevant.

Figure 3.11 shows example results of our method and TV in the FV4, FV6, HS1 and HS3 datasets. The leftmost column shows the input image. The second column is the result of our method. The third column is the saliency map result of TV. One can see that TV produces a very good quality saliency map. However, it can be affected by noisy input if the scale parameter is not set properly. Also, figures with different spacing between points obtain different saliency intensities, whereas in our method the NFA is invariant to the spacing. Finally, the fourth column presents the final curves extracted by hysteresis thresholding in the saliency map. Here again, setting a unique set of values for the high and low threshold that works well with and without the presence of noise is not trivial.

Figure 3.12 shows further examples. The order of the columns is the same as in Figure 3.11. The top two rows of Figure 3.12 show a masking by texture phenomenon. Our detector correctly determines that there is no salient structure in the second image. In the bottom row of Figure 3.12, our algorithm is able to find exactly the number of curves in the images. However, it prefers to split some of the curves in places where there are too strong changes in curvature. In fact, the triplet model described in Section 3.2 imposes a penalty on strong curvatures. In this case an iterative approach would possibly help finding the remaining pieces of curves. Exploring this issue remains as future work.

3.6 Conclusion

The work discussed in this chapter is an attempt to find a quantitative model for the grouping of dots under the good continuation gestalt. Our approach is, as in Chapter 2, a formalization of the non-accidentalness principle, based on a very simple model that favors local symmetries. This makes the model prefer smooth and long curves, where the dots are equally spaced. The model also accounts for the masking effect produced by surrounding noise points. We present an algorithm for the detection of good continuation groupings under this model. In the experimental section we presented through extensive examples the theoretical limits of the model as well as the results obtained with the algorithm. These informal experiments show a strong correlation between the quantitative measure introduced and human perception. Furthermore, we showed the superiority of the method with respect to the successful Tensor Voting approach in the figure-ground segmentation problem, constrained to dot patterns.

We envision three different lines of future work. One is the incorporation of the closure gestalt to the model [Elder and Zucker 1993, 1996]. The second is the utilization of more complex models for the local interactions, in line with local interaction fields studied in the literature [Field et al. 1993] and studying the particular case of junctions [Xia et al. 2014]. Third, the formulation of a coarse-to-fine version of the algorithm that would model a hierarchical process of perception and reduce the required computations [Pizlo et al. 1997].

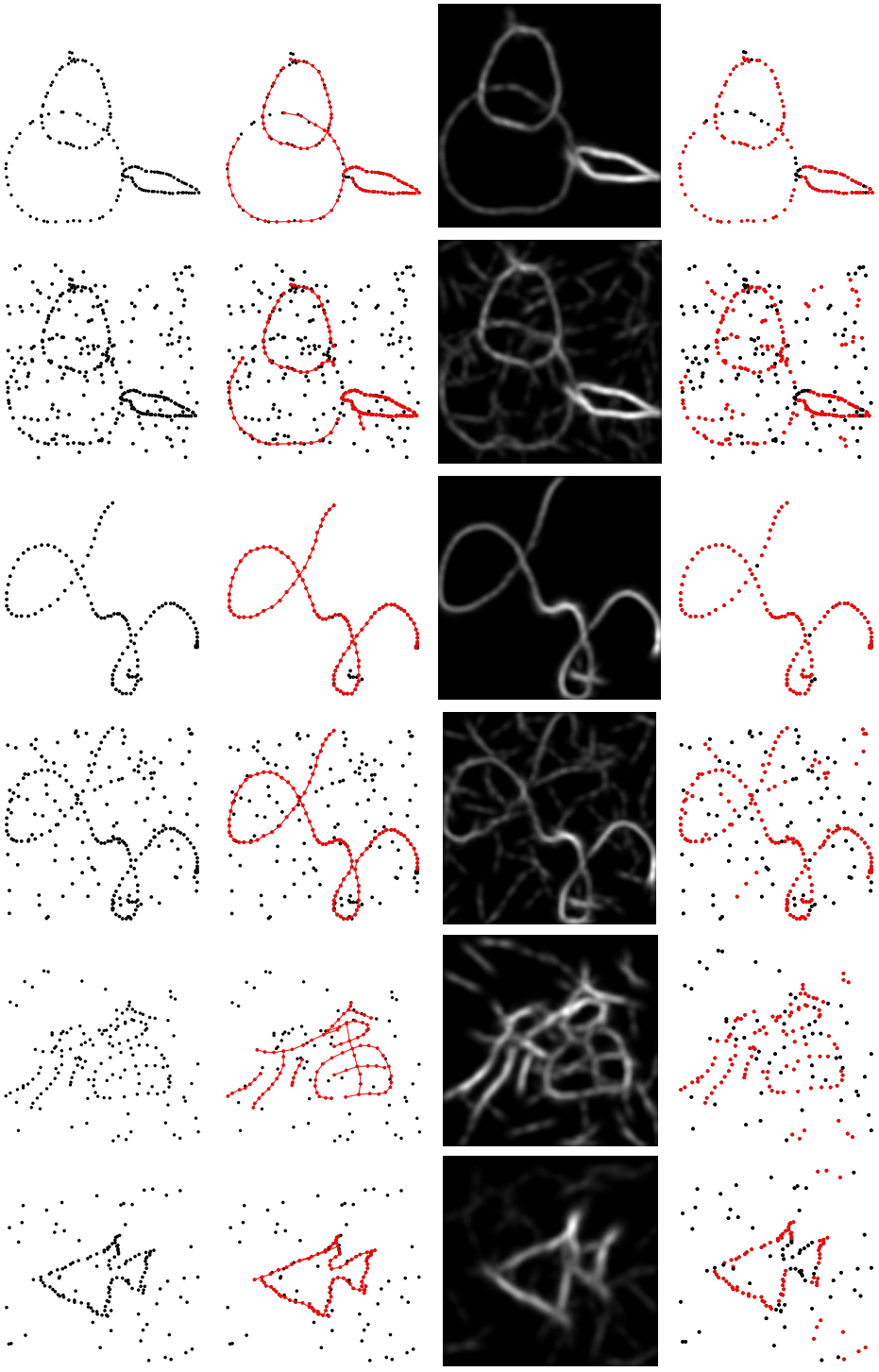


Figure 3.11: Results in example figures from the datasets described in Table 3.1. **Left column:** input points. From top to bottom: example extracted from dataset FV4, FV6, HS1 and HS3. **2nd column:** salient curves obtained with our method. **3rd column:** saliency map obtained by TV ($\sigma = 0.05$). **Right column:** points belonging to salient curves, extracted by thresholding with hysteresis the saliency map of TV. Note the difficulty of finding a set of parameters for TV that can distinguish structure from noise, but at the same time find structures at any scale.

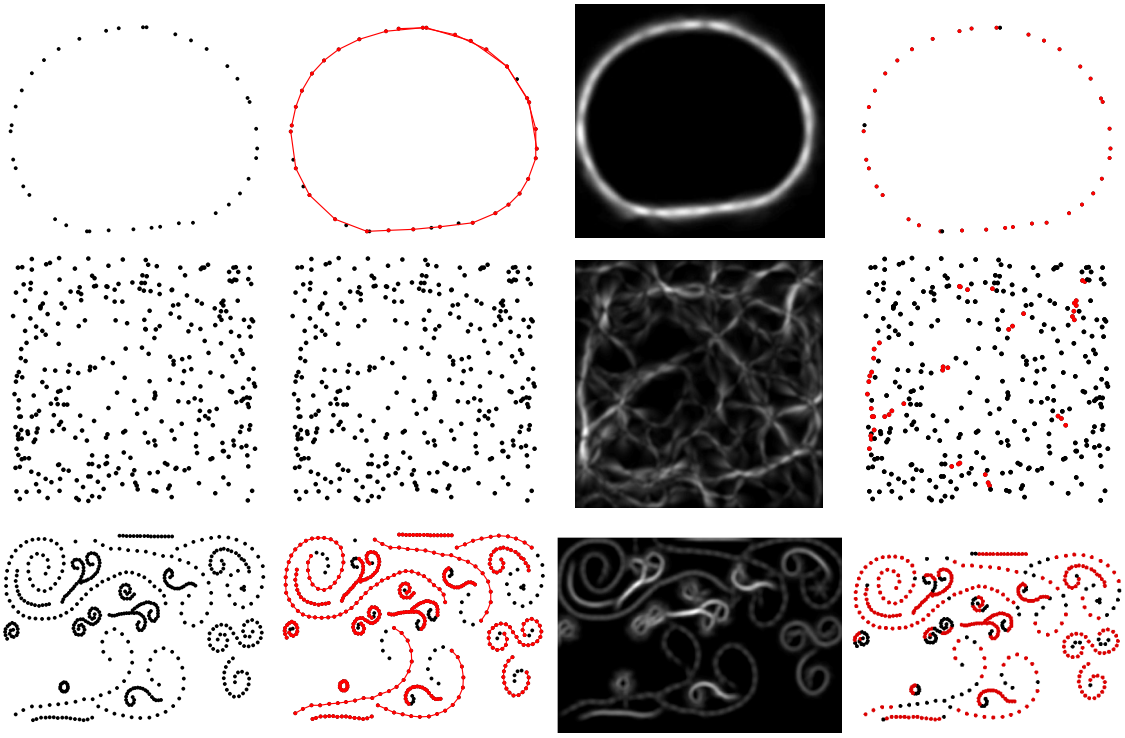


Figure 3.12: Additional results. **Left column:** input points. **2nd column:** result of our method. **3rd column:** saliency map obtained by TV ($\sigma = 0.05$). **Right column:** points belonging to salient curves, extracted by thresholding with hysteresis the saliency map of TV. **Top row:** example image from FV1. **2nd row:** the same image masked by 300 random points. Note how our method correctly determines that there are no perceptually relevant structures. TV is also able to resolve this case, although not without some spurious detections. Because the voting vectors are coming from many random locations, no distinct salient directions are obtained. **Bottom row:** image containing 21 curves at different scales. Our method correctly discovers the number of curves. However, one limitation of the method is shown: strong curvatures are penalized by the triplet model.

4 On Vanishing Points Detection

In this chapter we present a novel method for automatic vanishing point detection based on primal and dual point alignment detection. The point alignment detection algorithm described in Chapter 2 is used twice: First in the image domain to group line segment endpoints into more precise lines. Second, it is used in the dual domain where converging lines become aligned points. The use of the recently introduced PClines dual spaces and a robust point alignment detector leads to a very accurate algorithm. Experimental results on two public standard datasets show that our method significantly advances the state-of-the-art in the Manhattan world scenario, while producing state-of-the-art performances in non-Manhattan scenes.

The detection algorithm produced in this chapter can be tested online on any image at
`http://dev.ipol.im/~jlezama/ipol_demo/vanishing_points/`
(user: demo, password: demo)

4.1 Introduction

Under the pinhole camera model, 3D lines are transformed into 2D lines. Moreover, parallel lines in 3D are projected into lines that converge on a point (perhaps at infinity) known as a *vanishing point* (VP). In the presence of parallel lines, as is common in human-made environments, VPs provide crucial information about the 3D structure of the scene and have applications in camera calibration, single-view 3D scene reconstruction, autonomous navigation, and semantic scene parsing, to mention a few.

There is a vast literature on the problem of VP detection, starting with the seminal work by Barnard [1983] and leading to high-precision algorithms in recent works [Wildenauer and Hanbury 2012; Xu et al. 2013]. Typically, a method starts by the identification of oriented elements, which are then clustered into groups of concurrent directions, and finally refined to get the corresponding VP. Most proposed methods use image line segments as oriented elements [Barnard 1983; Collins and Weiss 1990; Almansa et al. 2003; Xu et al. 2013], but oriented edge points are also used [Coughlan and Yuille 2003; Denis et al. 2008; Barinova et al. 2010], or even the alignment of similar structures [Vedaldi and Zisserman 2012]. The clustering is often performed in the image plane, by identifying points or zones of concurrence of the elements [Rother 2002; Almansa et al. 2003]; other methods, however, rely on the Gaussian sphere of world directions [Barnard 1983; Collins and Weiss 1990], which requires camera calibration information. Less common is the use of a dual space in which points become lines and converging lines become aligned points [Ballard and Brown 1982; Zhao et al. 2013]. Various validation methods are used, from simple thresholds to statistical methods

[Collins and Weiss 1990; Almansa et al. 2003] or Bayesian inference [Coughlan and Yuille 2003; Denis et al. 2008]. Early methods used the Hough transform in the Gaussian sphere [Barnard 1983; Quan and Mohr 1989]. More recent methods rely on variations of RANSAC [Mirzaei and Roumeliotis 2011; Wildenauer and Hanbury 2012] or EM [Denis et al. 2008], which was successfully used to solve the uncalibrated case by Kosecká and Zhang [2002]. Some algorithms start with a clustering step that is obtained non-iteratively, and perform iterations to improve the result [Tardif 2009; Xu et al. 2013]. To simplify the often ill-posed problem, some methods make assumptions about the scene contents. The most common one is the so-called "Manhattan world", implying the existence of only three orthogonal VPs [Coughlan and Yuille 2003], which is sometimes inherently enforced during clustering [Mirzaei and Roumeliotis 2011; Bazin et al. 2012; Wildenauer and Hanbury 2012]. When valid, this assumption helps improving the results. Barinova et al. [2010], however, claim that a better balance between generality and robustness is provided by a relaxed assumption where one vertical VP and multiple horizontal ones (not necessarily orthogonal) are considered [Schindler and Dellaert 2004; Barinova et al. 2010]. Similar to Almansa et al. [2003], Tepper and Sapiro [2014b] is based on an *a contrario* model on the intersection of two lines in a region, except that instead of performing a convenient tiling of the image plane, the regions are given by solving an inverse pulley and belt problem. Recent works made progress by defining various consistency measures between VPs and line segments [Wildenauer and Hanbury 2012; Antunes and Barreto 2013; Xu et al. 2013], while Barinova et al. [2010] performs a joint optimization of line, VP and camera parameters.

In this work we build on the advances of various previous works to obtain a novel and more accurate VP detection algorithm. The oriented elements are the line segments detected with the LSD algorithm [Grompone von Gioi et al. 2012]. The clustering step is done in the dual space [Zhao et al. 2013], but takes advantage of the PClines point-to-line mappings described recently by Dubská et al. [2011]. The unsupervised point alignment detector of Chapter 2 is used to compute sets of collinear points, which correspond to converging lines and thus to VPs. The very same point alignment detector is used to group aligned line segments into longer and more precise ones. After the clustering is done and the candidate VPs are obtained, the NFA of the alignments is used to find the final triplet of VPs, when the Manhattan world assumption is applicable, or the horizon line when it is not. The method is deterministic and non-iterative and has an accuracy comparable or better than state-of-the-art algorithms.

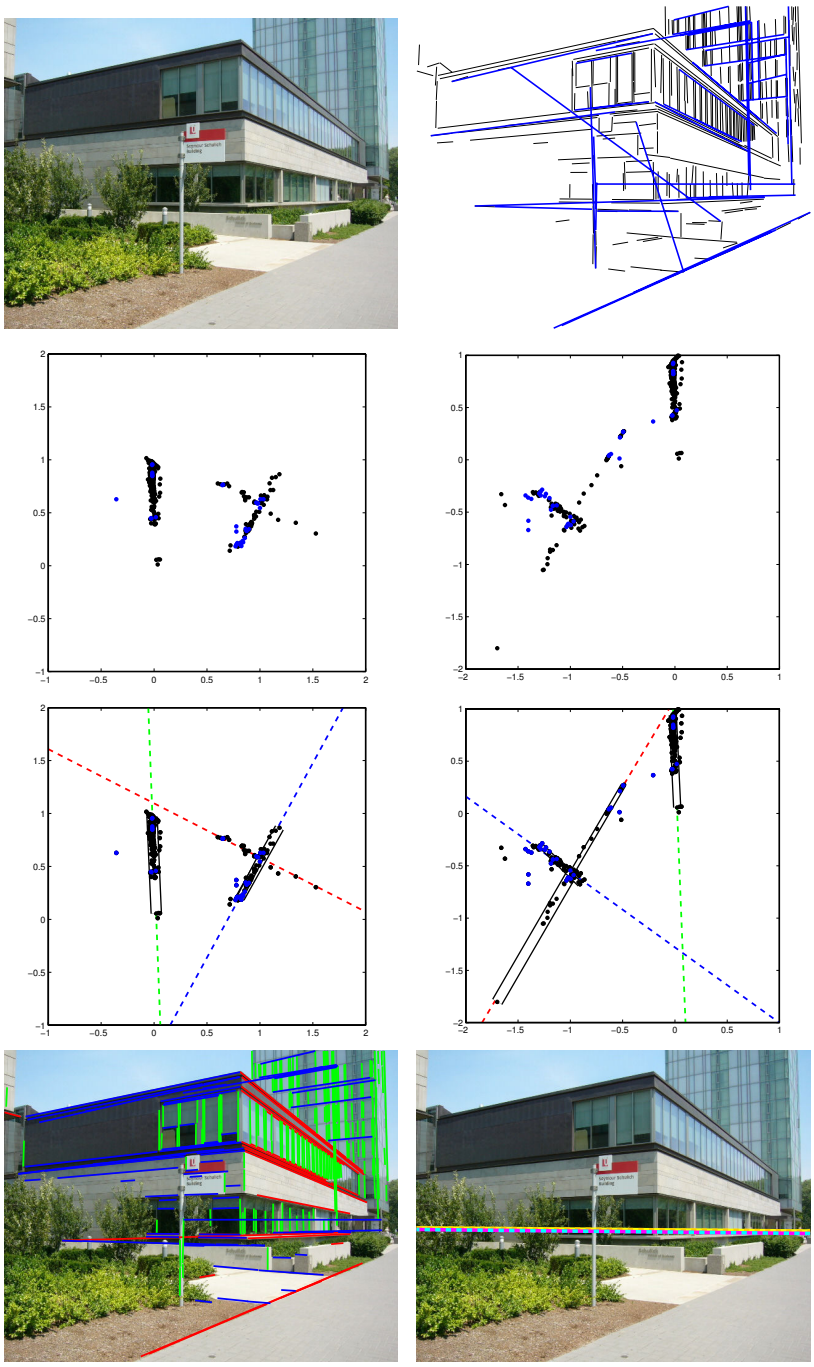


Figure 4.1: Main steps of our method. **Top-Left:** Input image. **Top-Right:** Line segments (black) and alignments of line segments endpoints (blue). **2nd Row:** Lines as points in *straight* (left) and *twisted* (right) PClines spaces. **3rd Row:** Aligned points detections (parallel black lines) and the ground truth (dashed lines). Some alignments are more visible in one of the spaces than in the other. **Bottom-Left:** Final VP associations by enforcing orthogonality. **Bottom-Right:** Horizon line estimation (yellow-orange: ground truth, cyan-magenta: ours).

4.2 Method Description

The proposed algorithm works in six steps, illustrated in Figure 4.1 and described in detail in the following subsections:

1. Detect line segments in the image using LSD [Grompone von Gioi et al. 2012].
2. Find and group aligned line segment endpoints using the unsupervised point alignment detector of Chapter 2*.
3. Transform line segments in the image into points in the PClines *straight* and *twisted* dual spaces Dubská et al. [2011]. In the PClines dual spaces, converging lines become aligned points. Two different dual spaces are used to cope with the unbounded nature of the mapping.
4. Detect point alignments in both dual spaces using the unsupervised point alignment detector of Chapter 2*. Detected point alignments, which correspond to converging line segments in the image, are considered as VP candidates.
5. Identify redundant detections (VPs detected in both dual spaces) and refine the position of the candidate VPs.
6. Identify relevant VPs and estimate the horizon line based on one of two hypothesis: Manhattan or non-Manhattan world.

* An optional accelerated version of this procedure is described in Section 2.8.

4.2.1 Point alignment detection

The proposed method uses the point alignment detector described in Chapter 2. In the VP detection procedure, this method will be used twice. First, to find alignments of segment endpoints, that can contribute to discover extra cues on vanishing directions. Second, to find alignments of points in the dual space, that correspond to converging line segments in the image. Note that the alignment detector does not require the number of clusters to be known in advance, so it can work with images with any number of VPs.

4.2.2 Segment endpoint alignments

The aim of this step is to exploit the alignment of features that a line segment detector alone cannot capture. For example, a regular arrangement of equally sized vertical poles along a road reveals a vanishing direction that would be unnoticed by the line segment detector, which would only discover the individual poles. A similar behavior with the vertical borders of windows can be seen in Figure 4.2. To exploit this type of cue, the point alignment detector of Chapter 2 is used to find alignments among the endpoints of the line segments detected by LSD [Grompone von Gioi et al. 2012]. In addition, this step increases the direction accuracy of short line segments by grouping them and brings an improvement to the VP detection performance[1].

First, line segments are grouped by length and orientation. The idea is that joinable line segments should be similar in length and orientation. This also allows to pass to the alignment detector small sets of points instead of the set of all the endpoints in the image, which significantly

[1] On average 2% with the metric used in Section 4.3.

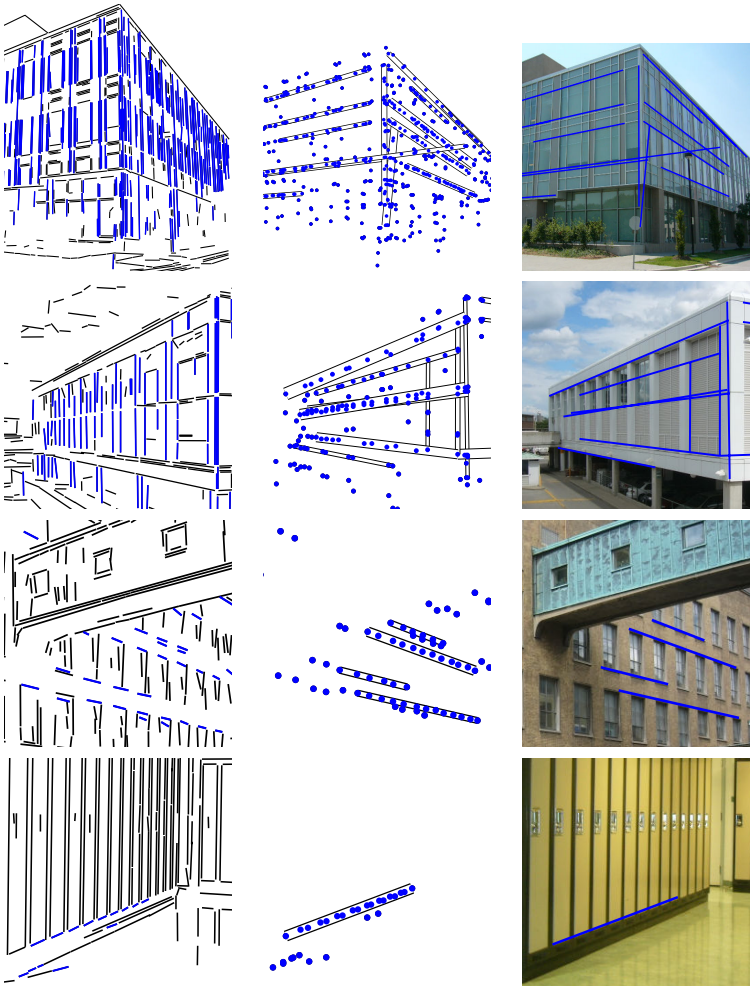


Figure 4.2: Examples of detected alignments of line segment endpoints. Best viewed in electronic format. **Left**: original line segments, highlighting in blue one group of length and orientation. **Center**: line segment endpoints and the detected alignments for that group (parallel black lines). **Right**: additional oriented features obtained by the method (blue lines). In the top two rows, parallel line segment endpoints are connected to recover structural directions that were not captured by the line segment detector alone. In the bottom rows, short, inaccurate line segments are joined into larger, more accurate ones.

reduces computation time. A single threshold τ is used on the length, and angular steps of 30° are used to group by orientation (with a 5° overlap between groups). The objective is to connect line segments that share the same orientation – e.g. the borders of the windows of a building – or endpoints of parallel line segments – e.g. an array of vertical structures. For each group of line segments, the point alignment detector is run over the line segment endpoints. This produces a new set of line segments. At the end of this stage, short (and therefore inaccurate) line segments are discarded. The final list of segments is composed of the long line segments and the line segments from the alignment of endpoints found among both the short and long ones. Figure 4.2 shows some examples of the grouping by length and orientation and the resulting detections. It can be seen that grouping the borders of windows creates long line segments that follow the direction of

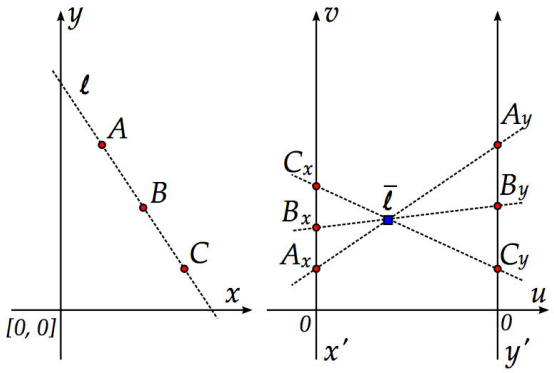


Figure 4.3: Schematic of the *straight* PClines transform. **Left**: three points and a line in the image $(x,\ y)$ space. **Right**: their representation in the PClines $(u,\ v)$ *straight* dual space. This figure was taken from Dubská et al. [2011].

the building structure. It is worth noting that some false or redundant detections are not harmful to the overall method, since they will only contribute to a few extra points in the dual space.

4.2.3 PClines parameterization

The problem of finding converging lines in the image can be posed as a point alignment detection problem by parameterizing the lines as points in a suitable dual space, where converging lines are mapped to aligned points. In such a space, the problem of VP detection has a simple interpretation: the detection of elongated clusters of points. Our method uses the PClines parameterization of Dubská et al. [2011], see Figure 4.3. The *straight* version uses a parallel coordinate system, where the horizontal x axis in the image is represented as the vertical v axis in the dual space, and the vertical y axis in the image is represented as a vertical line passing through $u = d$ in the dual space. Thus, a point $A = (A_x, A_y)$ in the image is represented in the dual space by a line passing by $(0, A_x)$ and (d, A_y). A line joining multiple points in the image is represented in the dual space as a point that lies at the intersection of the lines representing those points. Note that points in dual space can be arbitrarily far away from the origin Dubská et al. [2011]. To overcome this unboundedness problem, the *twisted* version of the PClines transform is also used, where the x axis stays in the ordinates axis but the $-y$ axis is transformed into a vertical line at $u = -d$. The second row of Figure 4.1 shows example results of the transformation for real data. The detailed algorithm is presented in Section B.4.

By using both the *straight* and *twisted* transforms, it is guaranteed that all VPs in the image are represented as a bounded set of points in at least one of the two spaces. The value of d is set to 1 and the limits of the dual domains where the point alignment detector will be executed are set to $[-1,\ 2] \times [-1,\ 2]$ for the *straight* transform and $[-2,\ 1] \times [-2,\ 1]$ for the *twisted* transform. Points that fall outside each domain are discarded.

Once the alignment detector is run in each dual space, each alignment found determines a candidate VP $\mathbf{v}_i$ in the image, with an associated meaningfulness NFA_i (Section 4.2.1). In the next step, VPs that were detected in both spaces are identified with a simple distance threshold δ and only the one with the best NFA is kept.

4.2.4 Refinement

Each rectangle candidate of the point alignment detector (Section 4.2.1) is defined by a pair of points. Thus, the direction it defines represents in the image the intersection of two lines, which forms an initial candidate VP, that needs to be refined. The refinement is done in the following way. Given a line segment $\mathbf{l}$ and a VP candidate $\mathbf{v}_i$, the consistency between $\mathbf{l}$ and $\mathbf{v}_i$ is considered as the angle between the direction of $\mathbf{l}$ and the ideal line passing through $\mathbf{v}_i$ and the midpoint of $\mathbf{l}$ (see Figure 4.4). This consistency measure is often used in the literature [Rother 2002; Denis et al. 2008]. The subset of line segments consistent with $\mathbf{v}_i$ is obtained by setting a threshold θ on this angle. Finally, the function for updating the VP estimate of Antunes and Barreto [2013] is used on this subset, which minimizes the weighted sum of the square of perpendicular distances from the VPs to the lines defined by the line segments. The weights are given by the segments lengths. Due to the good quality of the initial clusters, a single refinement iteration is enough. The implementation details of the candidates refinement is presented in Section B.6.

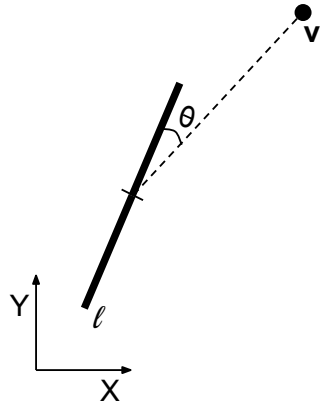


Figure 4.4: The angular error between a line segment $\mathbf{l}$ and a VP $\mathbf{v}$ is computed as the angle between the line segment direction and the line formed by the line segment midpoint and the VP.

4.2.5 Identifying relevant vanishing points

Once the refined candidate VPs are obtained, relevant detections must be discriminated from spurious ones. Some prior information on the relative positioning of the VPs can facilitate this task. We will consider two possible hypothesis.

The first one is the "Manhattan-world" hypothesis, which assumes that the image is dominated by three orthogonal VPs: one vertical and two horizontal. This assumption greatly simplifies the search for valid VPs, since it allows to discard VPs that do not comply with the orthogonality constraint and also to produce a third VP by imposing orthogonality when only two VPs are available.

The second, more relaxed hypothesis consists of considering one vertical VP and an unknown quantity of horizontal ones, that are not necessarily mutually orthogonal. In that case, the algorithm proceeds by identifying the vertical VP and based on the vertical VP it determines the horizontal ones. The procedure followed by the method under these two cases is detailed on the rest of this section.

In each case, using the obtained VPs, the method will estimate the horizon line in the image.

Manhattan world

When the camera parameters are known or have been correctly estimated, the candidate VPs can be represented in the Gaussian sphere. The Gaussian sphere is a unit sphere whose center is located at the center of the camera. A vanishing point in the image corresponds to a point in the Gaussian sphere (a 3D unitary vector), which is the intersection between the ray joining the camera center and the VP, and the surface of the sphere. To perform this conversion, the geometry of the camera needs to be known, in particular the principal point and the focal length in pixel units. The simple operations involved in this conversion are explained in detail in Section B.7. On what follows, when we refer to orthogonal VPs or to the angle between VPs, we will be referring to their representation in the Gaussian sphere. The representation space we are referring to will be clear in each context.

The "Manhattan-world" hypothesis assumes that the image contains three orthogonal VPs. To identify orthogonal triplets of VPs, the candidate vanishing points are converted to vectors in the Gaussian sphere, and orthogonality is determined based on a threshold on the angles between the vectors. Once the orthogonal triplets of VPs have been found, the one with the lowest sum of NFAs of the detections is selected as the valid VP triplet. When no orthogonal triplet is found, the most significant orthogonal pair is taken (in terms of summed NFA) and the third VP is obtained by the cross product of the two orthogonal vectors.

To compute the horizon line, the vertical VP is identified as the one with the largest absolute vertical coordinate. The remaining two VPs are the horizontal ones. The horizon line is simply the line passing through the horizontal VPs. Of course, this trivial solution requires the camera to be not far from horizontal with respect to the captured urban scene.

Non-Manhattan world

The Manhattan-world hypothesis is in general valid for simple urban scenes. However, for more complex scenes, this assumption is too strong and must be relaxed. A more relaxed hypothesis is the existence of one vertical and an unknown quantity of horizontal VPs, not necessarily mutually-orthogonal. We shall refer to this hypothesis as "non-Manhattan world". It is important to note that the unsupervised alignment detector used to compute candidate VPs in section 4.2.3 does not require the number of clusters to be known a priori, it is automatically computed. Thanks to this, the method can manage images with any number of VPs.

The procedure followed by the algorithm under this assumption is described next. It starts by identifying which of the candidate VPs is the vertical VP. Then, based on the estimated vertical VP and a set of constraints, it identifies the horizontal VPs.

To determine the vertical VP, the first step is to compute a set of vertical VP candidates, among all VPs. The vertical distance from a VP to the center of the image, and the angle with respect to the vertical axis of the image[2] are evaluated to form the subset of possible vertical VPs. Among this subset, the most meaningful VP (with lowest NFA) is kept as the zenith or vertical VP. Here again, in order for this heuristic to work the target scene is expected not to be heavily rotated with respect to the 3D horizontal plane.

The rest of the VPs are considered as candidates for horizontal VPs. To compute the final set of relevant horizontal VPs, these are filtered out based on two criteria: First, the VPs are converted to the Gaussian sphere and the VPs that are far from being orthogonal to the vertical VP are discarded. Note that to do this the focal length in pixel units f and the principal point must be known. However, these are used only to discard non-horizontal VPs, so an approximate value is generally

[2]We refer to the vertical axis as a vertical line passing through the image center.

sufficient. Second, the horizontal VPs that are obtained from clusters of parallel lines in the image – for which the problem of setting the horizon line height is undetermined – are also discarded, based on a distance threshold.

Because the horizon line is perpendicular to the line connecting the principal point and the vertical VP, the problem of estimating the horizon line from a set of candidate horizontal VPs is one-dimensional. The horizon line is obtained by a weighted vote, where each horizontal VP casts a vote and the weights are based on the NFA (Section 4.2.1) of each VP detection. The weight w_i for the VP $\mathbf{v}_i$ is:

$$w_i = \left(\frac{(-\log_{10} \mathrm{NFA}_i)}{\sum_j (-\log_{10} \mathrm{NFA}_j)} \right)^2 . \tag{4.1}$$

The implementation details of this procedure are described in Section B.7.

4.3 Experiments

In this experimental section we present example results of the method, as well a systematic performance evaluation on two standard public datasets for vanishing points detection.

4.3.1 Example results

For the experiments presented in this section, we used a C implementation of the point alignment detector and the C code of Grompone von Gioi et al. [2012] for line segment detection. The rest of the algorithm was implemented in MATLAB. Processing a 640x480 image takes an average of 22 seconds in a 2.4 Ghz Intel Core i5 laptop with 8 GB of RAM. The bottleneck of the method is the computation of point alignments in *straight* and *twisted* dual domains, which accounts for more than 90% of the processing time. This can be accelerated as explained in section 2.8.

Figures 4.5, 4.6 and 4.7 show example results of the method. The resulting image shows, on the top-right corner, the grouping of line segments in the image corresponding to each VP and the estimated horizon line. On the bottom rows, the PClines *straight* and *twisted* spaces are shown, with the point alignments that have been detected, and the representation of the final VPs. Figures 4.5 and 4.6 show urban scenes with a regular layout that adjusts well to the Manhattan-world hypothesis. In both cases, this hypothesis has been assumed. On Figure 4.7, the buildings are not arranged in a regular grid, and multiple horizontal VPs exist. By imposing the "non-Manhattan" world hypothesis, the algorithm finds multiple horizontal VPs. For further experimentation, the reader is invited to try the online demo of the method[3].

[3] `http://dev.ipol.im/~jlezama/ipol_demo/vanishing_points/`

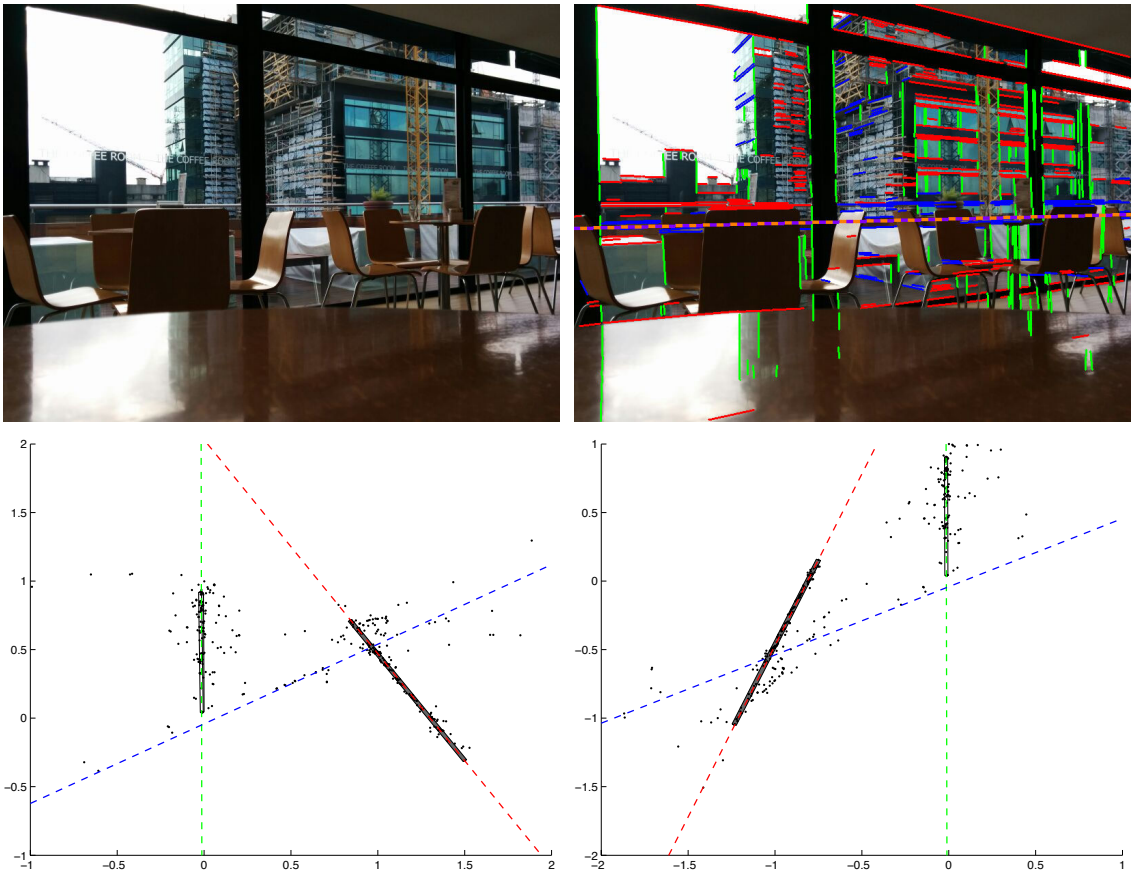


Figure 4.5: Example result of the method. Best viewed in electronic format. **Top left:** Original image. **Top right:** Algorithm result, line segments corresponding to each final VP detection and horizon line. **Bottom:** PClines *straight* (**left**) and *twisted* (**right**) dual spaces. Alignments (shadowed rectangles) and final VPs (colored dashed lines). In this case, the leftmost VP, represented by the blue dashed line and the blue segments in the result, was not found by the alignment detector in any of the dual spaces. However, it has been determined by applying the Manhattan-world hypothesis, which implies that the third VP must be orthogonal to the two VPs detected. For this example f was set to 0.9.

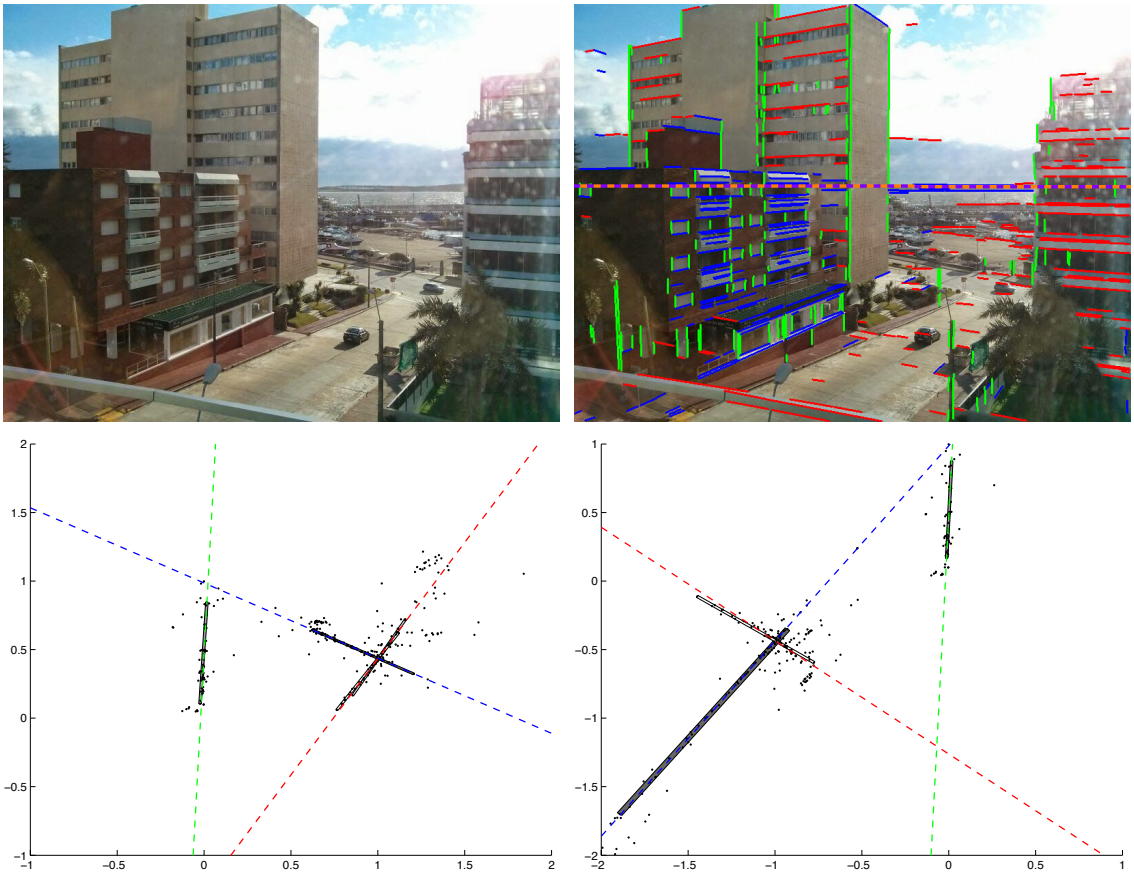


Figure 4.6: Example result of the method. Best viewed in electronic format. **Top left:** Original image. **Top right:** Algorithm result, line segments corresponding to each final VP detection and horizon line. **Bottom:** PClines *straight* (**left**) and *twisted* (**right**) dual spaces. Alignments (shadowed rectangles) and final VPs (colored dashed lines). In this case, the alignment detections corresponding to all VPs have been detected by the alignment detector. The difference in direction between the detections rectangles and the dashed lines shows that the refinement step has fine-tuned the final VP locations. For this example f was set to 0.95.

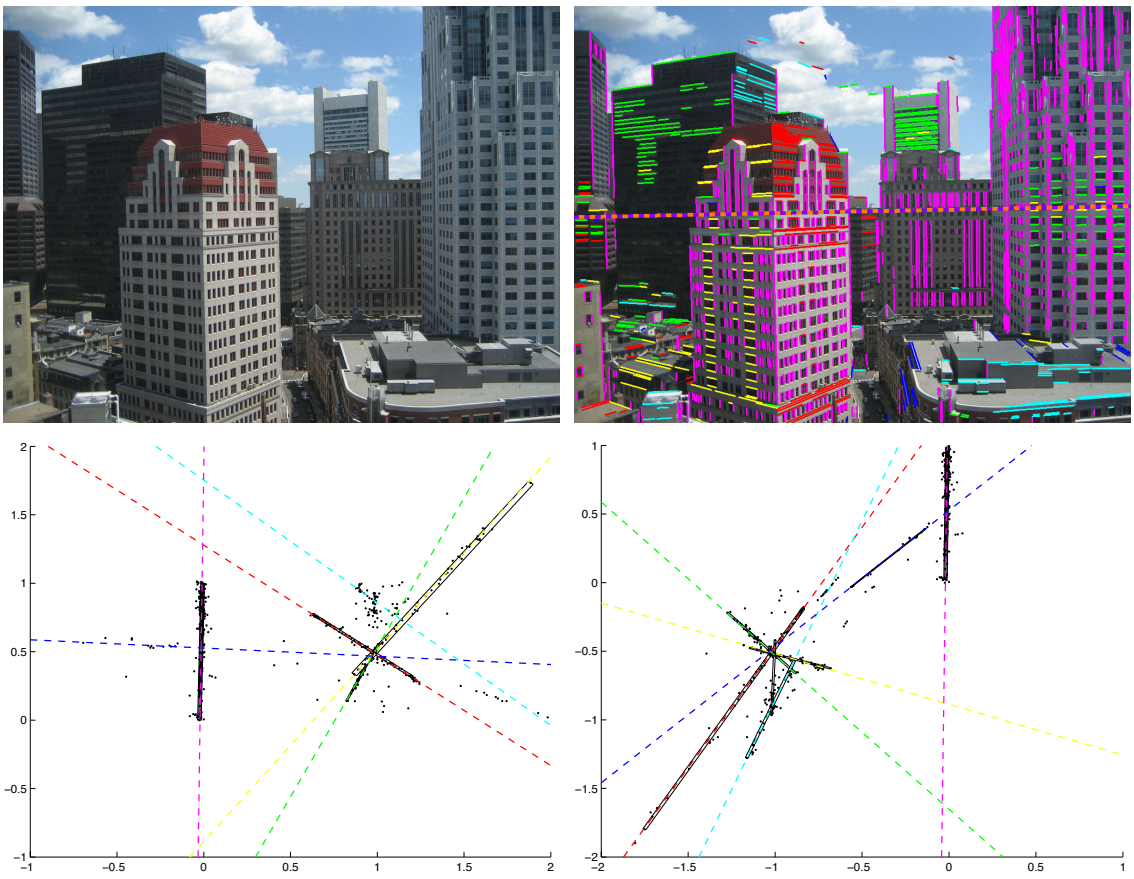
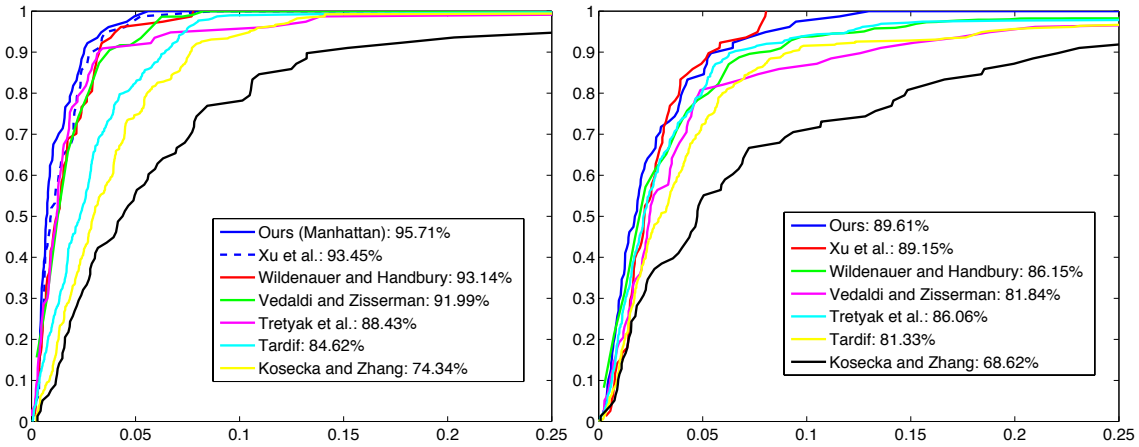


Figure 4.7: Example result of the method when the Manhattan-world hypothesis is not applicable. Best viewed in electronic format. **Top left:** Original image. **Top right:** Algorithm result, line segments corresponding to each final VP detection and horizon line. **Bottom:** PClines *straight* (**left**) and *twisted* (**right**) dual space. Alignments (shadowed rectangles) and final VPs (colored dashed lines). In this example multiple horizontal VPs have been found. On the dual space, one can see that the lines corresponding to the multiple horizontal VPs intersect close to a point which represents the horizon line in the image. The line in cyan is however a misdetection: it does not correspond to a true horizontal VP and is far away from the horizon line. For this example f was set to 1.



(a) York Urban Dataset (b) Eurasian Cities Dataset

Figure 4.8: Cumulative histograms of the horizon detection error. The horizontal axis represents the horizon line error [Barinova et al. 2010]. The vertical axis represents the ratio of images with horizon line error lower than the corresponding abscissa. Results for the competing methods have been kindly provided by Yiliang Xu.

4.3.2 Quantitative evaluation

A commonly used measure to assess the performance of VPs detection is the horizon line estimation error. This measure is defined as the maximum distance in the image domain between the estimated horizon line and the ground truth, divided by the image height. In this section, we use this measure to establish the performance of the method in two standard and widely used datasets.

The first dataset is the York Urban Dataset (YUD) [Denis et al. 2008]. It includes 102 images of outdoor and indoor scenes, the camera parameters, and the ground truth VPs. All the images satisfy the Manhattan world assumption, so the ground truth consists of 3 orthogonal VPs. Example results for this dataset are shown in Figure 4.11.

The second dataset is the Eurasian Cities Dataset (ECD) [Barinova et al. 2010], which presents a much more challenging dataset of 103 urban scenes that do not satisfy the Manhattan world assumption in general. They depict different architectural and urban styles and are taken by different cameras. Under these conditions, the method described in Section 4.2.5 is used. Example results for this dataset are shown in Figure 4.12.

Following the protocol of Barinova et al. [2010] and other recent works [Wildenauer and Han-

Manhattan-world assumption		No Manhattan-world assumption			
YUD *train*	YUD *test*	YUD *train*	YUD *test*	ECD *train*	ECD *test*
96.02%	95.71%	94.39%	94.32%	89.80%	89.61%

Table 4.1: Quantitative results of the algorithm as the area under the curve (AUC) of the horizon line estimation error. *Train* indicates the set of the first 25 images used to adjust the parameters. *Test* indicates the remaining images in the dataset. The third and fourth columns present the result of the algorithm when not applying the Manhattan-world on YUD (applying the exact same method as for ECD). At the moment of publication these results were state-of-the-art [Lezama et al. 2014a].

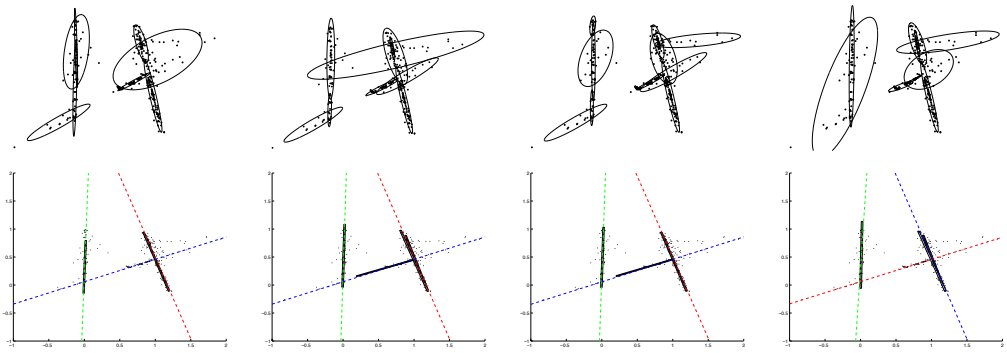


Figure 4.9: **Top:** Result of four different runs of the GMM routine in the PClines *straight* space for the image shown in Figure 4.10. Ellipses represent the obtained Gaussians. Each run of the GMM algorithm produces a different result, because of the random initialization. **Bottom:** Shadowed rectangles represent the alignments detections found using as candidate endpoints the extremes of the ellipses in the top row. Dashed lines represent the obtained VPs.

bury 2012; Vedaldi and Zisserman 2012; Xu et al. 2013], the first 25 images of each dataset are used to adjust the parameters of the method and the evaluation is performed on the remaining images. The performance score is measured as the area under the curve (AUC) of the cumulative histogram of the horizon line detection error. The performance scores obtained in both training and testing subsets for both datasets are reported in table 4.1. It also includes the result of applying the non-Manhattan version of the algorithm on YUD. Figure 4.8 shows the comparison of our results to those of Kosecká and Zhang [2002], Tardif [2009], Barinova et al. [2010], Vedaldi and Zisserman [2012], Wildenauer and Hanbury [2012] and Xu et al. [2013]. Table B.2 presents the values for the parameters, optimized by grid search on the first 25 images of each dataset.

4.3.3 Accelerated version

In this subsection we present an experimental analysis of the effects of using the accelerated version of the point alignment detector of Chapter 2 instead of the one based on exhaustive search. As pointed out in Section 2.8, the Gaussian mixtures algorithm of Figueiredo and Jain [2002] (which we will refer to as GMM) has a random initialization, which makes its result stochastic. Thus, the result of the accelerated VP detector, which depends on the result of the GMM procedure, will also be stochastic. This behavior is illustrated in Figure 4.9. Each run of GMM produces a different result. However, the most important clusters are usually captured, albeit with minor deviations. Once the ellipses shown in the top row of Figure 4.9 have been obtained, the alignment detector algorithm is run considering as possible endpoints for alignments, only the extremes of these ellipses. The bottom row of Figure 4.9 shows the detections thus obtained. One can see that the endpoints of the alignments correspond to extremes of the ellipses. Figure 4.10 shows the effect of the random result of GMM in the final estimation of the horizon line for an example image from YUD.

To obtain a more precise result, one may run the GMM procedure many times, producing a bigger list of ellipses and thus more candidates for point alignments. Naturally, as the algorithm is run more times, the probability of finding the good clusters increases, as does the computation time. An analysis of the cost in both performance and computation time of running multiple instances of the GMM procedure to generate alignment candidates is presented in Table 4.2 and 4.3. In Table 4.2, the same measure of performance used in Section 4.3.2, namely the horizon line

Figure 4.10: Example results of the accelerated version of the VP detection algorithm. Best viewed in electronic format. **Left:** Original image. **Center:** In red: Ground Truth horizon line. In blue: horizon lines obtained in each of the runs shown in Figure 4.9, superposed. Each run of the Gaussian Mixtures algorithm produces a different but similar result, because of the random initialization. **Right:** Detail of the green rectangle in the middle image.

estimation error, is used. The parameter k indicates the number of times the `GMM` procedure is run to produce candidate alignment endpoints. Since the algorithm is non-deterministic, it is run 10 times for each value of k, and the performance results are averaged. As expected, the more instances are run, the better the performance of the algorithm at a higher computation time cost. Table 4.3 presents the average computation time of the accelerated algorithm for one image, for each value of k.

Dataset	$k = 3$	$k = 6$	$k = 9$	$k = 12$	no acceleration
ECD	$82.32 \pm 1\%$	$82.71 \pm 1\%$	$84.33 \pm 1\%$	$85.69 \pm 1\%$	89.61%
YUD	$93.30 \pm 0.99\%$	$94.29 \pm 0.32\%$	$94.31 \pm 0.40\%$	$94.59 \pm 0.35\%$	95.71%

Table 4.2: Evaluation of the acceleration of the point alignment detection using the algorithm of Figueiredo and Jain [2002] to generate candidate alignments. The parameter k is the number of times the algorithm is run to create a list of possible clusters of aligned points. The performance shown is the AUC score as described in section 4.3.2 averaged on 10 runs.

Dataset	$k = 3$	$k = 6$	$k = 9$	$k = 12$	no acceleration
ECD	11.3 ± 4.1 s	23.0 ± 10.0 s	41.8 ± 20.0 s	65.4 ± 32.4 s	130.2 ± 105.6 s
YUD	5.0 ± 1.1 s	8.5 ± 3.3 s	13.5 ± 6.6 s	19.8 ± 10.9 s	23.7 ± 23.6 s

Table 4.3: Evaluation of the acceleration of the point alignment detection using the algorithm of Figueiredo and Jain [2002] to generate candidate alignments. k is the number of times the algorithm is run to create a list of possible clusters of aligned points. The time shown is the average computation time, evaluated on the whole dataset and divided by the number of images in it, averaged on 10 runs.

4.4 Conclusion

In this chapter we have presented a method for vanishing point detection based on four key ideas: First, the use of the robust point alignment detector described in Chapter 2, used in the image

domain as well as in dual space, without any modification or parameter tuning. Second, finding alignments of line segment endpoints, using again the point alignment detector, which enhances the accuracy of short line segments and produces new oriented elements as extra cues to the vanishing directions. Third, the use of the PClines parameterization in its two variants, *straight* and *twisted*, to improve the discriminability of vanishing points. Finally, exploiting the measure of meaningfulness provided by the alignment detector to estimate the horizon line. Our experimental results show that our method performs, in general, as well as state-of-the-art methods, and achieves significantly better accuracy when the Manhattan world assumption is applicable. We have also shown that an accelerated version of the method can be implemented, although at some cost in performance.

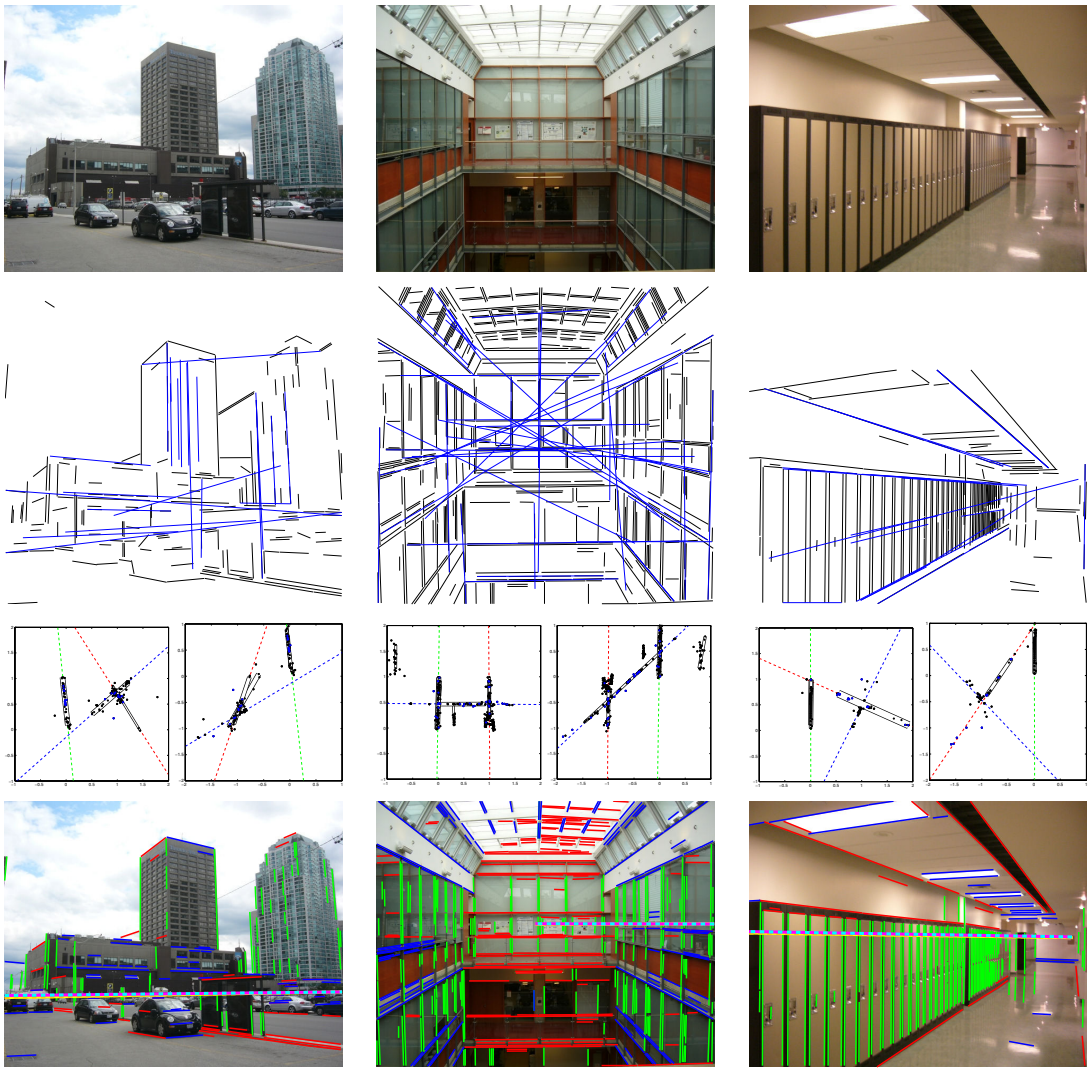


Figure 4.11: Example results of the method on selected images from YUD. **Top Row:** Original image. **2nd Row:** Line segments (black) and alignments of line segments endpoints (blue). **3rd Row:** PClines *straight* and *twisted* dual spaces and point alignment detections (parallel black lines). The ground truth is represented with colored dashed lines. **Bottom Row:** Line segments corresponding to each final vanishing point detection and horizon line (yellow-orange: ground truth, magenta-cyan: ours). In the last row, line segments from endpoint alignments have been removed for clarity. Note that the refinement and redundancy steps are not represented in this figure.

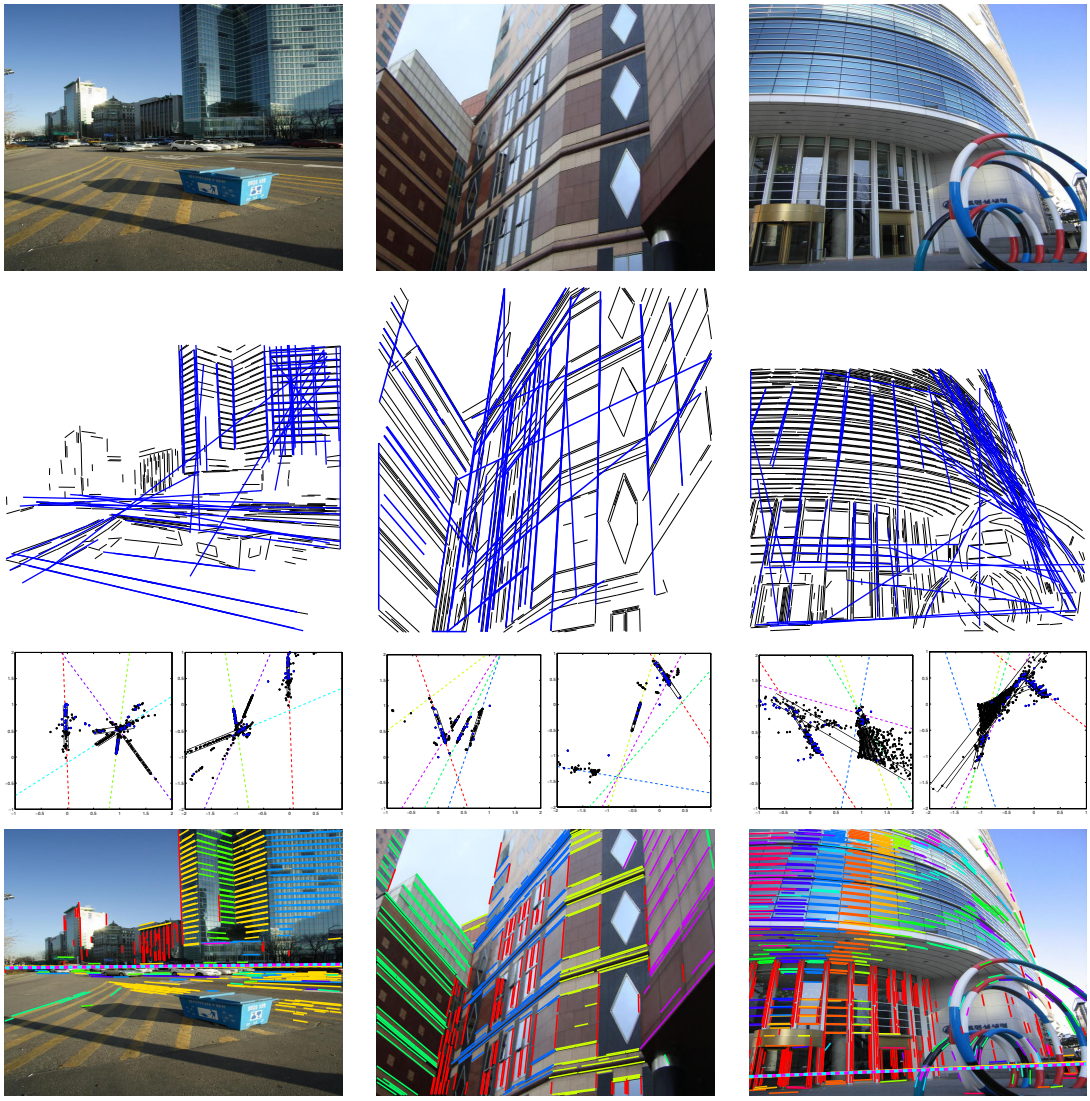


Figure 4.12: Example results of the method on selected images from ECD. **Top Row:** Original image. **2nd Row:** Line segments (black) and alignments of line segments endpoints (blue). **3rd Row:** PClines *straight* and *twisted* dual spaces and point alignment detections (parallel black lines). The ground truth is represented with colored dashed lines. **Bottom Row:** Line segments corresponding to each final vanishing point detection and horizon line (yellow-orange: ground truth, magenta-cyan: ours). In the last row, line segments from endpoint alignments have been removed for clarity. Note that the refinement and redundancy steps are not represented in this figure.

5 Conclusions and Perspectives

This dissertation concerns the quest for:

(i) mathematical models that describe the grouping principles of human perception

(ii) efficient algorithms based on such models that can also be applied to practical computer vision problems.

Our attempt to find mathematical models for perceptual grouping starts in a fairly constrained setting, the perception of structures in dot patterns. Although limited in complexity with respect to real images, dot patterns allow for rich visual interactions and trigger non-trivial mechanisms of human perception, most of which still remain uncharted territory for science.

Under this setting, we studied two types of structures. First, as a very basic case, we considered perceptually relevant alignments of dots. Secondly, we studied the more general case of grouping of dots by good continuation. In each case we observed the importance of dot density, regularity of spacing, length of the structure and the visual masking by the presence of random elements or other more perceptually relevant structures.

Based on these observations, we proposed for each of these two tasks a mathematical model and an algorithm whose quantitative prediction ability shows in general a strong correlation with human perception. Both models are based on evaluating the expectation of occurrence, under a random hypothesis, of dots falling in certain positions, given conveniently computed local estimations of density. The second model, naturally, uses a more flexible and general template that allows the search of long and smooth curves.

In this study we only advanced models for each partial gestalt but we did not address the competition between multiple different gestalts, or the problem of finding a global explanation of a figure, namely the *prägnanz* gestalt. The competition and collaboration of multiple gestalts remains a very important and extremely challenging open problem and a possible line of work for continuing this study. Figure 5.1 shows the limitations of using only partial gestalt detectors and the necessity of solving the competition and collaborations between them. Very recently, Tepper and Sapiro [2014a] presented a framework for finding multiple groupings by consensus, opening a very interesting perspective for this problem.

The study in the *a contrario* setting of other partial gestalts such as closure (which collaborates with good continuation)[Elder and Zucker 1993, 1996], symmetry and proximity (as generic clustering [Tepper et al. 2011]), to name a few, also remain widely open. It goes without saying that the partial gestalt algorithms presented in this dissertation leave room for refinement. In particular, improving their computation time is an important issue. To this end, a hierarchical or pyramidal analysis of the figure could be the key. It is natural to think that human perception does not compute the millions of possible combinations of the elements in a single pass, and some coarse to fine reasoning is likely to occur.

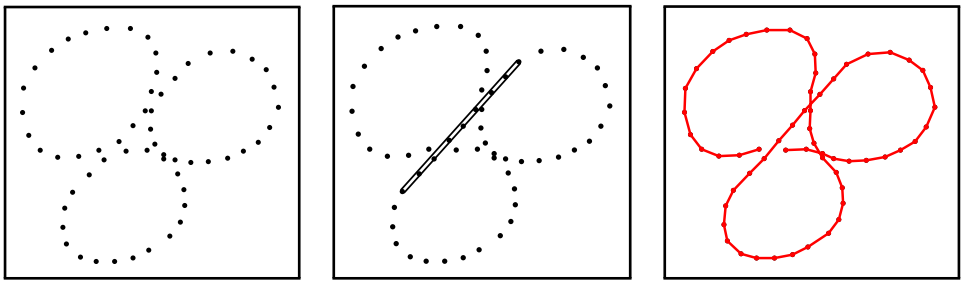


Figure 5.1: This figure shows the limitations of the partial gestalt detectors presented in this thesis. **(a)** input points. **(b)** result of the point alignment detector. **(c)** result of the good continuation detector. Finding a perceptually plausible global interpretation of this figure probably requires the competition of multiple gestalts: alignments, good continuation, closure and convexity. Modeling mathematically such a competition between grouping laws remains an open problem.

Human perception is able to quickly detect structures in dot patterns. Algorithms that are able to mimic human perception should be able to solve practical problems when these are presented in such a way that the solution is easily perceived by humans. We show such an example in the case of vanishing points detection, where the clustering solution in the dual space is easy to identify for a human observer. By using our perception inspired algorithm for this clustering problem we obtained state-of-the-art results, outperforming existing complex models of inference in the literature. Finding other practical problems where the solution can be perceived as a particular grouping of dots, and partial gestalt algorithms can be applied is also an interesting line of future work. Of course, with this we are not saying that the vanishing point detection problem is solved. There is room for improvement in our method as well.

A Detailed implementation of the point alignment detector

A.1 Main Algorithm

- Line 1: The purpose of this line is to create a data structure containing a symmetric reflection of the points in the input domain. This will be used when the local density estimation window falls outside the input domain. In that case, the points in the extension will be used for density estimation. Note that they will not be used to create candidate alignments.
- Line 2: Described in Algorithm 8
- Line 3: Described in Algorithm 9

A.2 Point Alignment Detector

Algorithm 8 describes the pseudo-code for the search and validation procedure. Following are some additional comments:

- Line 6: r is a rectangle whose main axis is the segment $\overline{\mathbf{x}_i\mathbf{x}_j}$ and whose width is w.
- Lines 9 and 10: R_1 and R_3 are depicted in Figure 2.5 (right).
- Line 11: The two points defining the statistical test must not be counted. In the present implementation, the counting of points inside the rectangles is based on the horizontal and vertical distances of each point to the segment $\overline{\mathbf{x}_i\mathbf{x}_j}$. A symmetric extension of the points is used when the local window falls outside the input domain. As a reference, the point counting procedure is done in the following way:

Algorithm 7: Main body of the algorithm

input : A set $\mathbf{X}$ of N points $[W = 8, L = 8, \mathcal{C} = \{1...\sqrt{N}\}, \varepsilon = 1]$
output: A list $\mathbf{m}$ of non-redundant point alignments

1 Create a symmetric extension of $\mathbf{X}$;
2 $\mathbf{l} =$ `detect_alignments`$(\mathbf{X})$;
3 $\mathbf{m} =$ `redundancy_reduction`$(\mathbf{l})$;

Algorithm 8: Point alignment detector

input : A set $\mathbf{X}$ of N points [$W = 8, L = 8, \mathcal{C} = \{1...\sqrt{N}\}, \varepsilon = 1$]
output: A list $\mathbf{l}$ of point alignments

for $i = 1$ *to* N **do**
  for $j = 1$ *to* $i - 1$ **do**
    $l \leftarrow$ distance$(\mathbf{x}_i, \mathbf{x}_j)$;
    $w \leftarrow l/10$;
    for 1 *to* W **do**
      $r \leftarrow$ rect$(\mathbf{x}_i, \mathbf{x}_j, w)$;
      $w_L \leftarrow l$; `// width of the local window`
      for 1 *to* L **do**
        $R_1 \leftarrow$ local-win-left$(\mathbf{x}_i, \mathbf{x}_j, w_L)$;
        $R_3 \leftarrow$ local-win-right$(\mathbf{x}_i, \mathbf{x}_j, w_L)$;
        Count M_1, M_2, M_3 the number of points in R_1, r and R_3 respectively;
        Compute $n^*(R, \mathbf{X})$ [eq.2.7];
        for $c \in \mathcal{C}$ **do**
          Divide r into c equal boxes;
          Compute $p_1(r, R, c)$ [eq.2.9];
          Count $b(r, c, \mathbf{X})$, the number of occupied boxes;
          Compute NFA$(r, R, c, \mathbf{X})$ [eq.2.12];
          if NFA$(r, R, c, \mathbf{X}) \leq \varepsilon$ **then**
            $\mathbf{l} \leftarrow r$;
        $w_L \leftarrow w_L/\sqrt{2}$;
      $w \leftarrow w/\sqrt{2}$;

```
m1 = 0;
m2 = 0;
m3 = 0;

dx = xj-xi;
dy = yj-yi;
len = sqrt(dx*dx+dy*dy);

for(m=0; m<N; m++)
{
  if( m==i || m==j ) continue; /* do not count i and j */

  /* compute coordinates relative to alignments */
  x =  dx * (points[2*m]-x1) + dy * (points[2*m+1]-y1);
  y = -dy * (points[2*m]-x1) + dx * (points[2*m+1]-y1);

  /* count local window points */
```

```
    if( x < 0.0 || x >= len ) continue; // hor. outside R
    if( y < -l/2.0 || y > l/2.0 ) continue; // ver. outside R
    if( y < -w/2.0 ) ++m1; //R_1 count
    else if( y > w/2.0 ) ++m3; // R_3 count
    else ++m2;
  }
```

Where `(xi,yi)` and `(xj,yj)` are the coordinates of the pair of points $\mathbf{x}_i$ and $\mathbf{x}_j$ forming the alignment, `N` is the total number of points, `w` and `l` are the widths of the rectangle and local window respectively. `points` is an array of length $2N$ containing all points coordinates, such that `points[2*m]` and `points[2*m+1]` are the horizontal and vertical coordinates of the m[th] point respectively. Please note that this code is a simplified version of the source code, so variable names in the source code might differ.

- Line 13: Recall that $\mathcal{C}$ contains numbers from 1 to $\sqrt{N}$. However, in the present implementation, once the number of points inside the alignment is known, the algorithm starts from $c = n/2$, where n is the number of points inside the alignment and goes up to $c = 2n$. This is only an acceleration heuristic and does not affect the detector performance.

- Line 15: Note that n^*, the conservative estimation of points in R, is required to compute p_0.

- Line 16: When the rectangle is divided into c boxes, two half-boxes are left at each extreme of the alignment, so that c full boxes remain in the middle (see Figure A.1). There are two reasons for this. One is that the points defining the statistical test must not be counted in the test. The second reason is that if there are c points perfectly spaced inside the alignment, then these would fall exactly in the center of each of the c boxes. As a reference, the present implementation counts the number of occupied boxes in the following way:

```
box = len / (c+1); // takes into account
                   // one half-box at each extreme

for(m=0; m<n_alignment; m++)
{
  b = floor( (x[m] - box/2.0) / box );
  if( b>=0 && b<c ) ++occupied[b];
}
```

Where `len` is the length of the alignment, `n_alignment` is the number of points inside the alignment, `x` is an array of length `n_alignment` containing horizontal distances inside the alignment and `occupied` is an array of length `c` initialized with zeros that will contain a positive integer where a box is occupied or zero otherwise. Please note that this code is a simplified version of the source code, so variable names in the source code might differ.

- Line 17: To simplify the computation of the NFA, eq. 2.12, our implementation computes a bound to the tail of the binomial distribution instead of its actual value. Given n independent Bernoulli random variables with parameter p, its sum $S = X_1 + \cdots + X_n$ follows the binomial distribution. It is easy to obtain Desolneux et al. [2008] an upper bound to the tail of this distribution using Hoeffding's inequality Hoeffding [1963]:

$$\mathcal{B}(n,k,p) = \mathbb{P}(S \geq k) \leq \left(\frac{p}{k/n}\right)^k \left(\frac{1-p}{1-k/n}\right)^{n-k}.$$

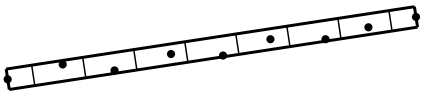


Figure A.1: A detail of the counting of occupied boxes. Two half boxes are placed at the extremes of the rectangle, and only the full boxes in the middle count. This is because the two points forming the alignment must not be counted in the statistical test since they are the ones that define the test.

Algorithm 9: Redundancy reduction (Masking Principle)

input : A list $\mathbf{l}$ of all significant alignments
output: A list $\mathbf{m}$ of maximally significant alignments

1 $\mathbf{l} \leftarrow$ sort($\mathbf{l}$); `// by NFA, lowest to highest`
2 $\mathbf{m}[0] \leftarrow \mathbf{l}[0]$;
3 **for** $i = 1 \rightarrow \text{length}(\mathbf{l}) - 1$ **do**
4 $B = \mathbf{l}[i]$;
5 masked = false;
6 **for** $j = 0 \rightarrow \text{length}(\mathbf{m}) - 1$ **do**
7 $A = \mathbf{m}[j]$;
8 $\mathbf{X}' = \{\mathbf{x} \mid \mathbf{x} \notin r_A\}$;
9 **if** $\text{NFA}(r_B, R_B, c_B, \mathbf{X}') > \varepsilon$ *[eq.2.12]* **then**
10 masked = true ;
11 **break**;
12 **if** masked $==$ false **then**
13 $\mathbf{m} \Leftarrow \mathbf{l}[i]$;

The following code computes the previous approximation and returns the logarithm with base 10 of the result:

```
static double log_bin(int n, int k, double p)
{
  double r = (double) k / (double) n;
  if( r <= p ) return 0.0;
  else if( n == k) return k * log10(p);
  else return k * log10(p/r) + (n-k) * log10( (1-p)/(1-r) );
}
```

A.3 Redundancy Reduction

Algorithm 9 presents the pseudo-code for the redundancy reduction procedure. Following are some additional comments:

- Lines 4 and 7: Event A is defined by the rectangle r_A, local window R_A and the number

of boxes c_A. Event B is defined by the rectangle r_B, local window R_B and the number of boxes c_B.

- Line 8: A new set of points $\mathbf{X}'$ is considered, by removing all points belonging to alignment A (all points in r_A).
- Line 10: The NFA of B is re-evaluated using the set of points $\mathbf{X}'$. If the new NFA is greater than ε, then B is no longer significant when the points from A are removed: A masks B.
- Line 12: This condition means that the current alignment was not masked by any event in the current output list (all of which are more significant than itself). If the condition is satisfied, the current alignment is appended to the output list.

B Detailed implementation of the vanishing point detector

B.1 Main Algorithm

Algorithm 10 presents the pseudo code for the main vanishing point (VP) detection procedure. Each subroutine is detailed and explained in the rest of this section. Parameters are detailed in each subroutine and in table B.1. The following notation will be used:

- bold lowercase letters for vectors (e.g. $\mathbf{v}$)
- bold uppercase letters for lists (e.g. $\mathbf{L}$)
- uppercase letters for matrices (e.g. Q).
- $x_{\mathbf{v}}$ and $y_{\mathbf{v}}$ for the horizontal and vertical coordinates of a point $\mathbf{v}$ in the image, respectively.

The input to the algorithm is the image I. The output is a list $\mathbf{V}$ of relevant VPs locations, and the estimated horizon line $\mathbf{l}_{hor}$. The method has ten parameters. Parameters f and $\mathbf{p}$ are given by the geometry of the camera. The threshold τ is applied on the length of line segments, and is relative to the image size. Thresholds θ, δ and ζ are used to refine the position of the candidate VPs and remove redundant detections. Parameters γ_S and γ_R are angular parameters that are used to determine orthogonality or non-orthogonality in the Gaussian sphere. The parameters ω and λ are thresholds used to discard or validate candidate VPs. The use of each parameter will be detailed in the section corresponding to each subroutine.

Lines 1 and 2 perform the line segment detection and denoising (grouping and discarding short line segments). Lines 3 to 5 perform the mapping of the line segments into points in the dual spaces, and the detection of alignments in them. Lines 6 to 8 convert the alignments found in the dual space to points in the image, which are the candidate VPs. The procedure in line 9 refines the position in the image of each candidate VP. In line 10, the algorithm is divided into two possible paths: assuming the Manhattan-world hypothesis or not.

B.2 Line Segment Detection

The `LSD` method in line 1 of Algorithm 10 runs the Line Segment Detector of Grompone von Gioi et al. [2012] on the image I. The output is a list of line segments identified by their endpoints $((x_1, y_1), (x_2, y_2))$.

Algorithm 10: Main body of the algorithm

input : An image I. Parameters f, $\mathbf{p}$, τ, θ, ζ, δ, γ_S, γ_R, ω, λ (see table B.1).
output: A list $\mathbf{V}$ of detected VPs and $\mathbf{l}$, the estimated horizon line.

1 $\mathbf{L} \leftarrow \texttt{LSD}(I)$
2 $\mathbf{L} \leftarrow \texttt{denoise\_lines}(\mathbf{L}, \tau)$ // Algorithm 11
3 $\mathbf{X}_{straight}, \mathbf{X}_{twisted} \leftarrow \texttt{pclines\_transform}(\mathbf{L})$ // Algorithm 12
4 $\mathbf{A}_{straight} \leftarrow \texttt{detect\_alignments}(\mathbf{X}_{straight})$ // Algorithm 14
5 $\mathbf{A}_{twisted} \leftarrow \texttt{detect\_alignments}(\mathbf{X}_{twisted})$
6 $\mathbf{V}_{straight} \leftarrow \texttt{inverse\_pclines\_transform}(\mathbf{A}_{straight})$ // Algorithm 13
7 $\mathbf{V}_{twisted} \leftarrow \texttt{inverse\_pclines\_transform}(\mathbf{A}_{twisted})$
8 $\mathbf{V} \leftarrow \mathbf{V}_{straight} \bigcup \mathbf{V}_{twisted}$
9 $\mathbf{V} \leftarrow \texttt{refine\_detections}(\mathbf{V}, \theta, \delta, \zeta)$ // Algorithm 15
10 **if** *Manhattan-world* **then**
11 $\quad \mathbf{V}, \mathbf{l}_{hor} \leftarrow \texttt{manhattan\_constraints}(\mathbf{V}, f, \mathbf{p}, \gamma_S)$ // Algorithm 18
12 **else**
13 $\quad \mathbf{V}, \mathbf{l}_{hor} \leftarrow \texttt{non-manhattan\_constraints}(\mathbf{V}, f, \mathbf{p}, \gamma_R, \omega, \lambda)$ // Algorithm 23

B.3 Line Segment Denoising

The aim of this step is to remove short, noisy line segments, and to capture the direction of higher level structures formed by groupings of line segment endpoints. Algorithm 11 presents the pseudo code for this procedure. To achieve its goal, this procedure uses the alignment detector of Chapter 2 to find alignments of line segment endpoints. The line segments are grouped by length and orientation. First, they are divided into "short" and "long" segments using a threshold τ. Then, they are subdivided by orientation in angular slots of 40° with 5° overlap.

The alignment detection algorithm is run on the endpoints of the line segments contained in each slot (lines 5 to 7 and lines 9 to 11). The list of final line segments is composed by the long segments and the segments found as endpoint alignments (lines 3, 8 and 12). The benefits of this stage are twofold: First, noisy short segments are removed. Second, additional cues on vanishing directions are obtained when parallel line segments are grouped.

B.4 Dual space transformation

To map lines in the image into points in a dual space, the method uses the PCLines transformation Dubská et al. [2011]. The fundamentals of this transformation are presented in Dubská et al. [2011], Section 2. The simple operations required to compute the mapping into the *straight* and *twisted* spaces are described in Algorithm 12 for the direct transform (image to dual space) and in Algorithm 13 for its inverse (dual space to image). Note that in the direct transform, an output domain is fixed, and points falling outside it are discarded (Algorithm 12, lines 14, 16). This is because the output points will be passed to the alignment detector, which can only operate on a restricted domain (Section B.5). The domains are fixed to $[-1, 2] \times [-1, 2]$ in the *straight* space and $[-2, 1] \times [-2, 1]$ in the *twisted* space.

By using both *straight* and *twisted* spaces, it is guaranteed that each group of converging line

Algorithm 11: `denoise_lines`: Line segments denoising

input : A list of line segments $\mathbf{L}$. A length threshold τ.
output: A list $\mathbf{L}'$ of denoised line segments.

1 $\mathbf{L}_{short} \leftarrow \{l : l \in \mathbf{L}, \texttt{length}(l) \leq \tau\}$
2 $\mathbf{L}_{long} \leftarrow \{l : l \in \mathbf{L}, \texttt{length}(l) > \tau\}$
3 $\mathbf{L}' \leftarrow \mathbf{L}_{long}$
4 **for** $\alpha \in \{0°, 30°, 60°, 90°, 120°, 150°\}$ **do**
5 $\quad \mathbf{L}^{\alpha}_{short} \leftarrow \{l : l \in \mathbf{L}_{\mathbf{short}}, \alpha - 20° \leq \texttt{ang}(l) < \alpha + 20°\}$
6 $\quad \mathbf{x}^{\alpha}_{short} \leftarrow \texttt{endpoints}(\mathbf{L}^{\alpha}_{short})$
7 $\quad \mathbf{A}^{\alpha}_{short} \leftarrow \texttt{detect\_alignments}(\mathbf{x}^{\alpha}_{short})$ `// Algorithm 14`
8 $\quad$ append $\mathbf{A}^{\alpha}_{short}$ to $\mathbf{L}'$
9 $\quad \mathbf{L}^{\alpha}_{long} \leftarrow \{l : l \in \mathbf{L}_{\mathbf{long}}, \alpha - 20° \leq \texttt{ang}(l) < \alpha + 20°\}$
10 $\quad \mathbf{x}^{\alpha}_{long} \leftarrow \texttt{endpoints}(\mathbf{L}^{\alpha}_{long})$
11 $\quad \mathbf{A}^{\alpha}_{long} \leftarrow \texttt{detect\_alignments}(\mathbf{x}^{\alpha}_{long})$ `// Algorithm 14`
12 $\quad$ append $\mathbf{A}^{\alpha}_{long}$ to $\mathbf{L}'$

segments in the image will appear in at least one of the dual spaces as a bounded set of points. A simple analysis of the transformation shows that lines with an orientation close to 45° are mapped into points close to infinity in the *straight* transform, but are bounded for the *twisted* version. On the contrary, lines with an orientation of -45° are mapped to infinity in the *twisted* transform, but not in the *straight* one.

B.5 Alignment detection in dual space

To find clusters of aligned points in the dual space, the method uses the unsupervised alignment detector of Chapter 2. A relaxed threshold of $\varepsilon = 10$ in the number of false alarms is always used (it allows in theory some 10 false alarms per image). Since future steps in the algorithm will refine relevant detections and remove spurious ones, false positives are not a serious problem. Furthermore, the relaxed detection threshold can help by finding alignments at the limit of detection. The exact same version of the algorithm and its parameters is used in Algorithm 10, lines 4 and 5 and in Algorithm 11, lines 7 and 11. Algorithm 14 presents a short pseudo code for this procedure. Note that the domain of the input points must be defined and passed to the algorithm. When working in the image, as in Algorithm 11, the domain is the same as the image domain ($[0, W] \times [0, H]$). When working in the *straight* and *twisted* dual spaces, the domains are $[-1, 2] \times [-1, 2]$ and $[-2, 1] \times [-2, 1]$ respectively, as explained in Section B.4. For clarity, this has not been explicited in the calls to `detect_alignments` in Algorithms 10 and 11.

Lines corresponding to alignments detected in the dual space are converted to points in the image space by using the inverse PCLines transform described in Algorithm 13.

B.6 Vanishing point detections refinement

The point alignments detections in both PCLines dual spaces constitute the first candidates for VPs (recall that lines in the dual space correspond to points in the image). Because of the nature of the

Algorithm 12: `pclines_transform`: PCLines transformation of line segments in the image into points in dual space.

input : A list of line segments $\mathbf{L}$, $d = 1$. Height and width of the image, H and W.
output: Two lists $\mathbf{X}_{straight}$ and $\mathbf{X}_{twisted}$ of points in the dual space.

1 $\mathbf{X}_{straight} \leftarrow \emptyset$
2 $\mathbf{X}_{twisted} \leftarrow \emptyset$
3 **for** $\mathbf{l} \in \mathbf{L}$ **do**
4 $\quad (x_1, y_1), (x_2, y_2) \leftarrow$ `endpoints(l)`
5 $\quad x_1 \leftarrow \frac{x_1}{W}, y_1 \leftarrow \frac{y_1}{H}, x_2 \leftarrow \frac{x_2}{W}, y_2 \leftarrow \frac{y_2}{H}$
6 $\quad d_y \leftarrow y_2 - y_1$
7 $\quad d_x \leftarrow x_2 - x_1$
8 $\quad m \leftarrow \frac{d_y}{d_x}$
9 $\quad b \leftarrow \frac{(y_1 \cdot x_2 - y_2 \cdot x_1)}{d_x}$
10 $\quad u_{straight} \leftarrow \frac{d}{(1-m)}$
11 $\quad v_{straight} \leftarrow \frac{b}{(1-m)}$
12 $\quad u_{twisted} \leftarrow \frac{-d}{(1+m)}$
13 $\quad v_{twisted} \leftarrow \frac{-b}{(1+m)}$
14 $\quad$ **if** $(u_{straight}, v_{straight}) \in [-1, 2] \times [-1, 2]$ **then**
15 $\quad\quad$ Append $(u_{straight}, v_{straight})$ to $\mathbf{X}_{straight}$
16 $\quad$ **if** $(u_{twisted}, v_{twisted}) \in [-2, 1] \times [-2, 1]$ **then**
17 $\quad\quad$ Append $(u_{twisted}, v_{twisted})$ to $\mathbf{X}_{twisted}$

point alignment detector, the direction of the detected alignments is given by the two points at the extremes. This direction is of course biased and needs to be refined. The refinement procedure consists of two parts, as described in Algorithm 15. First, the candidate VPs are refined using the line segment information in the image domain. This process is described in Algorithm 16. Second, the set of candidate VPs is searched for redundant detections. This process is detailed in Algorithm 17.

To refine a candidate VP the following two steps are taken. First, the group of line segments converging to the candidate VP is searched for. This search is done only among the line segments detected by LSD (not the ones obtained as alignments of endpoints in Section B.3). The angle between a line segment and the line passing by the midpoint of the line segment and the candidate VP is considered (see Figure 4.4). Thresholding this angle, the list of line segments converging to the candidate VP is obtained (Algorithm 16, lines 3 to 8). Considering this subset of line segments, the position of the candidate VP is refined by using the VP update rule of Antunes and Barreto [2013]. This update rule finds the point in the image minimizing the weighted sum of the geometric distances between the VP and the line segments. We kindly refer the reader to Antunes and Barreto [2013] for the detailed development of the minimization problem and only give the resulting formula here. The update process is described in lines 9 to 21 of Algorithm 16.

Given N segments $\mathbf{l}_i = [a_i, b_i, c_i]^T$, the following 3x3 matrix Q, which is derived from the

Algorithm 13: `inverse_pclines_transform`: Inverse transformation of lines in PCLines space to points in the image

input : A list of lines $\mathbf{L} = \{\mathbf{l} : v = m \cdot u + b\}$. The PCLines space identifier (*straight* or *twisted*). Parameter $d = 1$. Height and width of the image, H and W.

output: A list of points $\mathbf{X} = \{(x, y)\}$ in the image domain.

1 $\mathbf{X} \leftarrow \emptyset$
2 **for** $\mathbf{l} : v = m \cdot u + b \in \mathbf{L}$ **do**
3 $x \leftarrow b$
4 **if** *straight* **then**
5 $y \leftarrow d \cdot m + b$
6 **else if** *twisted* **then**
7 $y \leftarrow -(-d \cdot m + b)$
8 Append $(x \cdot W, y \cdot H)$ to $\mathbf{X}$

Algorithm 14: `detect_alignments`: Point alignments detection.

input : A list of points $\mathbf{X}$, a rectangular domain D, the Number of False Alarms threshold $\varepsilon = 10$.

output: A list $\mathbf{A}$ of alignment detection.

1 $\mathbf{A} \leftarrow$ `detect_alignments`$(\mathbf{X}, D, \varepsilon)$

formula of the orthogonal distance is computed,

$$Q = \sum_{i=1}^{N} w_i^2 \frac{\mathbf{l_i} \mathbf{l_i}^T}{\mathbf{l_i}^T I_s \mathbf{l_i}} \tag{B.1}$$

where $I_s = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$ and w_i are the normalized lengths of the line segments, so the longest line segment has length 1. The estimated vanishing point $\mathbf{v}$ is given, in homogeneous coordinates, by the solution of the right nullspace problem:

$$[2Q,\ -\mathbf{p}] \begin{pmatrix} \mathbf{v} \\ \lambda \end{pmatrix} = \mathbf{0}, \tag{B.2}$$

where $\mathbf{p} = (0\ 0\ 1)^T$ (Algorithm 16, line 15). Note that $[2Q,\ -\mathbf{p}]$ is a 3x4 matrix and $\begin{pmatrix} \mathbf{v} \\ \lambda \end{pmatrix}$ is a 4x1 vector, because $\mathbf{v}$ is in homogeneous coordinates. In MATLAB, the function `null()` is used to obtain the result.

It could happen that the matrix Q is ill conditioned, for example when the line segments in the group are mostly parallel. The method detects ill conditioned cases by measuring the normalized distance between the original VP and the refined VP. If it is above a certain threshold ζ, the original VP is kept (Algorithm 16, lines 18 to 21).

Algorithm 15: `refine_detections`: Post-processing of alignment detections.

input : A list of candidate VPs $\mathbf{V}$. An angle threshold θ and distance thresholds δ and ζ

output: A list $\mathbf{V}'$ of refined candidate VPs.

1 $\mathbf{V}' \leftarrow$ `refine_vps`$(\mathbf{V}, \mathbf{L}, \theta, \zeta)$; `// Algorithm 16`

2 $\mathbf{V}' \leftarrow$ `remove_redundancy`$(\mathbf{V}', \delta)$; `// Algorithm 17`

The second step in the refinement is the removal of the redundant detections, described in Algorithm 17. These redundancies can have two causes. One type is caused by detections corresponding to the same VP found both in the *straight* and *twisted spaces*. The alignment detection algorithm can also produce some redundant detections when a noisy cluster of aligned points triggers multiple thin detections. The identification of redundant detections is done by running single-link agglomerative clustering on the VP candidates in the image (Algorithm 17, line 3). The single-link agglomerative clustering is a common clustering procedure and we shall only briefly describe it. In the single-link agglomerative clustering, each VP starts in a cluster of its own. The clusters are then sequentially combined into larger clusters. At each step, the two clusters separated by the shortest distance are combined. The shortest distance is the minimum distance between two elements (one in each cluster). The algorithm stops when the shortest distance is above a threshold δ. For the detection of redundant VPs, the distances used are normalized, so that

$$d(\mathbf{v}_i, \mathbf{v}_j) = \frac{||\mathbf{v}_i - \mathbf{v}_j||}{\max(||\mathbf{v}_i, \mathbf{v}_j||)}. \tag{B.3}$$

For each cluster found, only the VP with lowest NFA (most significant) is kept (line 5).

Algorithm 16: `refine_vps`: Refine VP detections.

input : A list $\mathbf{V}$ of candidate VPs. A list $\mathbf{L}$ of line segments. An angle threshold θ and a distance threshold ζ.
output: A list $\mathbf{V}'$ of refined candidate VPs.

$\mathbf{V}' \leftarrow \emptyset$
for $\mathbf{v} \in \mathbf{V}$ **do**
  $\mathbf{L}' \leftarrow \emptyset$
  for $\mathbf{l} \in \mathbf{L}$ **do**
    $\mathbf{m} \leftarrow \texttt{mid\_point}(\mathbf{l})$
    $\mathbf{l}' \leftarrow \overline{\mathbf{mv}}$
    if $\mathit{angle}(\mathbf{l}, \mathbf{l}') < \theta$ **then**
      Append $\mathbf{l}$ to $\mathbf{L}'$
  $Q \leftarrow 3 \times 3$ matrix with all zeroes
  $I \leftarrow 3 \times 3$ identity matrix
  $M \leftarrow \max(\texttt{lengths}(\mathbf{L}'))$
  for $\mathbf{l} \in \mathbf{L}'$ **do**
    $w \leftarrow \texttt{length}(\mathbf{l})/M$
    $Q \leftarrow Q + w^2 \cdot \frac{\mathbf{l}\mathbf{l}^T}{\mathbf{l}^T I \mathbf{l}}$
  $\mathbf{p} \leftarrow [0, 0, 1]^T$
  $A \leftarrow [2Q,\ -\mathbf{p}]$
  $\mathbf{v}' \leftarrow \texttt{null\_space}(A)$
  if $\frac{||\mathbf{v}-\mathbf{v}'||}{||\mathbf{v}||} < \zeta$ **then**
    Append $\mathbf{v}'$ to $\mathbf{V}'$
  else
    Append $\mathbf{v}$ to $\mathbf{V}'$; `// Reject ill-conditioned case.`

Algorithm 17: `remove_redundancy`: Remove redundant VP detections.

input : A lists of VP candidates, $\mathbf{V}$. A distance threshold δ
output: A single list $\mathbf{V}$ of non-redundant VP candidates.

$\mathbf{V} \leftarrow \mathbf{V}_{straight} \bigcup \mathbf{V}_{twisted}$
$\mathbf{V}' \leftarrow \emptyset$
$\mathcal{C} \leftarrow \texttt{single\_link\_agglomerative\_clustering}(\mathbf{V}, \delta)$
for $\mathbf{C} \in \mathcal{C}$ **do**
  $\mathbf{v}' \leftarrow \arg\min_{\mathbf{v} \in \mathbf{C}} \mathrm{NFA}(\mathbf{v})$
  Append $\mathbf{v}'$ to $\mathbf{V}'$

B.7 Orthogonality Constraints and Horizon Line estimation

Once a list of refined candidate VPs is obtained, the last part of the VP detection algorithm consists in selecting the subset of VPs that best correspond to the real VPs in the image. To relax this problem, two possible hypothesis are introduced. One is the "Manhattan-world" hypothesis, which implies that only three orthogonal VPs are present in the image. The second one, which we will refer to as "non-Manhattan-world" is the existence of one vertical and multiple, non necessarily mutually-orthogonal horizontal VPs. The final step of the algorithm has two variants, one for each hypothesis. The two variants are described in the rest of this section.

Manhattan World

Algorithm 18 presents the pseudo code for the procedure followed by the method when the Manhattan-world hypothesis can be assumed. The procedure starts by converting the candidate VPs in the image to unit vectors in the Gaussian sphere using the `image_to_gaussian_sphere` routine described in Algorithm 21. To do this, it is necessary to know or estimate the focal length in pixel units and the position of the principal point or image center. After this conversion, triplets of orthogonal VPs are searched for using the `orthogonal_triplets` routine described in Algorithm 19 and the triplet with the lowest combined NFA is selected as the final result (Algorithm 18, line 8). (Recall that each candidate VP has an NFA associated which is the NFA of the corresponding alignment detected in the dual space). When no orthogonal triplets are found (lines 9 to 14), the most significant orthogonal pair is taken using the procedure in Algorithm 20, and the third vector is obtained by the cross product, and then refined using the `refine_vps` procedure of Algorithm 16.

In line 17, the VPs are be converted back to image coordinates using the `gaussian_sphere_to_image` procedure described in Algorithm 22. In lines 19 to 21 the vertical VP is identified as the one with the largest vertical coordinate and the remaining two horizontal VPs are used to trace the horizon line.

Non-Manhattan World

We refer to a "non-Manhattan" world scenario as a particular relaxation of the simple "Manhattan" world model. This relaxed hypothesis assumes the image has one vertical VP and multiple horizontal VPs, which are orthogonal to the vertical VP, but not necessarily orthogonal between them. Examples of these scenes can be seen in Figure 4.12. Under this assumption, the method starts by estimating which of the candidate VPs is the vertical one, and then selecting a possible set of horizontal VPs. This procedure is detailed in Algorithm 23.

In line 1 of Algorithm 23, the `vertical_vp_candidates` routine, detailed in Algorithm 24, is used to select a subset of the VPs that are candidates for the vertical VP. This routine is based on two thresholds on the image coordinates of the VPs. Given a candidate VP $\mathbf{v}$, its vertical coordinate $y_{\mathbf{v}}$ must be outside the image domain (larger than H), and a threshold ω is used for the angle between a vertical line passing through the image center, ($\mathbf{l}_{ver}$ in line 1 of Algorithm 24) and the line connecting the image center with the candidate VP ($\mathbf{l}_{\mathbf{v}}$ in line 5 of Algorithm 24). If both conditions are met the VP is added to the list of candidates for the vertical VP (line 6 of Algorithm 24). Of course, this is a trivial procedure and it would fail if the image is heavily rotated, but it works well for approximately horizontal urban scenes, for which the algorithm is intended. Finally,

among all the vertical VP candidates, the one with the lowest NFA is chosen (Algorithm 23, line 2).

Once the vertical VP has been determined the rest of the VPs are considered as horizontal VP candidates (line 3). Then, two lists are created. The first one, $\mathbf{V}_{hor}^{ortho}$ (line 5), will contain the horizontal VP candidates that are orthogonal to the vertical VP. The second list, $\mathbf{V}_{hor}^{finite}$ (line 10) will contain the horizontal candidate VPs that do not lie at "infinity".

The computation of the list of VPs orthogonal to the vertical VP, $\mathbf{V}_{hor}^{ortho}$, is described in lines 4 to 9 of Algorithm 23. The VPs are converted to the Gaussian sphere, based on known or estimated values for the focal length in pixel units, and the principal point. Using a threshold γ_R on the angle between vectors in the Gaussian sphere, the VPs that are far from being orthogonal to the vertical VP are discarded (lines 6 to 9).

The computation of the list of non-infinite VPs, $\mathbf{V}_{hor}^{finite}$, is described in lines 10 to 15. The aim is to identify VPs caused by sets of parallel lines, whose vertical position is undetermined. This is done with a threshold λ on the distance between the VP and the image center $\mathbf{p}$ (lines 11 to 13). This threshold is proportional to the image size (see table B.1). Note that it could happen that no VP is finite under this criterion. In that case the result is given by taking the VP closest to the image center (lines 14 to 15).

The list of final horizontal VPs is the intersection of the two aforementioned lists, namely those VPs that are both orthogonal to the vertical VP and finite. In the extreme case that the lists are disjoint, only one horizontal VP is considered, the one with the lowest NFA. This is described in lines 16 to 19.

Once the final subset of horizontal VP candidates is determined, the horizon line is obtained by a weighted average: each candidate horizontal VP casts a vote for the horizon line. This procedure is detailed in Algorithm 25. For each VP, its proposed horizon line is the line passing by itself, perpendicular to the line formed by the image center and the vertical VP (line 5). Since all lines are parallel, this problem is actually one-dimensional: what is being determined is the height of the horizon line, h_{hor} (line 2). The weight of each vote is given by the NFA associated to the VP, more precisely as $-log_{10}(\mathrm{NFA})$, normalized over the sum of this value for all remaining VPs (line 4).

Algorithm 18: `manhattan_constraints`: Final VP and horizon line estimation under the Manhattan-world hypothesis.

input : A list of candidate VPs in image domain $\mathbf{V}$. A threshold γ_S. f, the focal length in pixel units. $\mathbf{p}$, the principal point.

output: A list $\mathbf{V}$ of candidate VPs in image domain. The horizon line $\mathbf{l}_{hor}$.

$\mathbf{V}_u \leftarrow \emptyset$

for $\mathbf{v} \in \mathbf{V}$ **do**

  $\mathbf{v}_u \leftarrow$ `image_to_gaussian_sphere(`$\mathbf{v}, f, \mathbf{p}$`)` `// Algorithm 21`

  Append $\mathbf{v}_u$ to $\mathbf{V}_u$

$\mathbf{T} \leftarrow$ `orthogonal_triplets(`$\mathbf{V_u}, \gamma_S$`)` `// Algorithm 19`

$\mathbf{P} \leftarrow$ `orthogonal_pairs(`$\mathbf{V_u}, \gamma_S$`)` `// Algorithm 20`

if $\mathbf{T} \neq \emptyset$ **then**

  $\{\mathbf{v}_u^{(1)}, \mathbf{v}_u^{(2)}, \mathbf{v}_u^{(3)}\} \leftarrow \arg\min_{\mathbf{t} \in \mathbf{T}} \big(\text{score}(\mathbf{t})\big)$

else

  `// No orthogonal triplet found, use pairs`

  $\{\mathbf{v}_u^{(1)}, \mathbf{v}_u^{(2)}\} \leftarrow \arg\min_{\mathbf{p} \in \mathbf{P}} \big(\text{score}(\mathbf{p})\big)$

  $\mathbf{v}_u^{(3)} \leftarrow \mathbf{v}_u^{(1)} \times \mathbf{v}_u^{(2)}$

  $\mathbf{v}_{img}^{(3)} \leftarrow$ `gaussian_sphere_to_image(`$\mathbf{v}_u^{(3)}, f, \mathbf{p}$`)` `// Algorithm 22`

  $\mathbf{v}_{img}^{(3)} \leftarrow$ `refine_vps(`$\mathbf{v}_{img}^{(3)}$`)` `// Algorithm 16`

  $\mathbf{v}_u^{(3)} \leftarrow$ `image_to_gaussian_sphere(`$\mathbf{v}_{img}^{(3)}, f, \mathbf{p}$`)`

$\mathbf{V} \leftarrow \emptyset$

for $i \in \{1, 2, 3\}$ **do**

  $\mathbf{v}_{img}^{(i)} \leftarrow$ `gaussian_sphere_to_image(`$\mathbf{v}_u^{(1)}, f, \mathbf{p}$`)`

  Append $\mathbf{v}_{img}^{(i)}$ to $\mathbf{V}$

$\mathbf{v}^{ver} \leftarrow \arg\max_{\mathbf{v} \in \mathbf{V}} (y_{\mathbf{v}})$

$\{\mathbf{v}^{hor1}, \mathbf{v}^{hor2}\} \leftarrow \mathbf{V} \setminus \mathbf{v}^{ver}$

$\mathbf{l}_{hor} \leftarrow \overline{\mathbf{v}^{hor1}\mathbf{v}^{hor2}}$

Algorithm 19: `orthogonal_triplets`

input : A list of unit norm vectors $\mathbf{V}_u$ (with associated NFAs). A threshold γ_S.
output: A list $\mathbf{T}$ of orthogonal triplets of vectors.

$\mathbf{T} \leftarrow \emptyset$
for $\mathbf{v}_u^{(i)} \in \mathbf{V}_u$ **do**
  for $\mathbf{v}_u^{(j)} \in \mathbf{V}_u \setminus \mathbf{v}_u^{(i)}$ **do**
    for $\mathbf{v}_u^{(k)} \in \mathbf{V}_u \setminus \mathbf{v}_u^{(i)}, \mathbf{v}_u^{(j)}$ **do**
      if $\max(|\mathbf{v}_u^{(i)} \cdot \mathbf{v}_u^{(j)}|, |\mathbf{v}_u^{(i)} \cdot \mathbf{v}_u^{(k)}|, |\mathbf{v}_u^{(j)} \cdot \mathbf{v}_u^{(k)}|) < \gamma_S$ **then**
        $\mathbf{t}^{(i,j,k)} \leftarrow \left(\mathbf{v}_u^{(i)}, \mathbf{v}_u^{(j)}, \mathbf{v}_u^{(k)}\right)$
        $\text{score}(\mathbf{t}^{(i,j,k)}) = \text{NFA}(\mathbf{v}_u^{(i)}) + \text{NFA}(\mathbf{v}_u^{(j)}) + \text{NFA}(\mathbf{v}_u^{(k)})$
        Append $\mathbf{t}^{(i,j,k)}$ to $\mathbf{T}$

Algorithm 20: `orthogonal_pairs`

input : A list of unit norm vectors $\mathbf{V}_u$ (with associated NFAs). A threshold γ_S.
output: A list $\mathbf{P}$ of orthogonal pairs of vectors.

$\mathbf{T} \leftarrow \emptyset$
for $\mathbf{v}_u^{(i)} \in \mathbf{V}_u$ **do**
  for $\mathbf{v}_u^{(j)} \in \mathbf{V}_u \setminus \mathbf{v}_u^{(i)}$ **do**
    if $|\mathbf{v}_u^{(i)} \cdot \mathbf{v}_u^{(j)}| < \gamma_S$ **then**
      $\mathbf{p}^{(i,j)} \leftarrow \left(\mathbf{v}_u^{(i)}, \mathbf{v}_u^{(j)}\right)$
      $\text{score}(\mathbf{p}^{(i,j)}) = \text{NFA}(\mathbf{v}_u^{(i)}) + \text{NFA}(\mathbf{v}_u^{(j)})$
      Append $\mathbf{p}^{(i,j)}$ to $\mathbf{P}$

Algorithm 21: `image_to_gaussian_sphere` : Convert VP candidates in the image to unit vectors in the Gaussian sphere

input : A VP in image coordinates $(x_\mathbf{v}, y_\mathbf{v})$. The focal ratio f in pixel units and the principal point $\mathbf{p}$.
output: The corresponding unit-norm vector $\mathbf{v}_u$ in the Gaussian sphere.

$\mathbf{v}_u \leftarrow \left(x_\mathbf{v} - x_\mathbf{p},\ \ y_\mathbf{v} - y_\mathbf{p},\ \ f\right)^T$
$\mathbf{v}_u \leftarrow \frac{\mathbf{v}_u}{||\mathbf{v}_u||}$

Algorithm 22: `gaussian_sphere_to_image` : Convert VP candidates as unit vectors in the Gaussian sphere to points in the image.

input : A unit-length vector $\mathbf{v}_u$ in the Gaussian sphere. The focal ratio f in pixel units and the principal point $\mathbf{p}$.

output: The corresponding VP in image coordinates $\mathbf{v}_{img}$

$\mathbf{v}_{img} \leftarrow \left(f \cdot \frac{\mathbf{v}_u(1)}{\mathbf{v}_u(3)} + x_{\mathbf{p}}, f \cdot \frac{\mathbf{v}_u(2)}{\mathbf{v}_u(3)} + y_{\mathbf{p}}\right)^T$

Algorithm 23: `non-manhattan_constraints`: Final VP and horizon line estimation without the Manhattan-world hypothesis.

input : A list of candidate VPs in image domain $\mathbf{V}$. f, the focal length in pixel units. $\mathbf{p}$, the principal point. Thresholds γ_R, ω, λ

output: A list $\mathbf{V}$ of candidate VPs. The horizon line $\mathbf{l}_{hor}$.

$\mathbf{V}_{ver} \leftarrow$ `vertical_vp_candidates`$(\mathbf{v})$ `// Algorithm 24`

$\mathbf{v}^{(v)} \leftarrow \underset{\mathbf{v} \in \mathbf{V}_{ver}}{\arg\min}\big(\mathrm{NFA}(\mathbf{v})\big)$

$\mathbf{V}_{hor} \leftarrow \mathbf{V} \setminus \mathbf{V}_{ver}$

$\mathbf{v}_u^{(v)} \leftarrow$ `image_to_gaussian_sphere`$(\mathbf{v}^{(v)}, f, \mathbf{p})$ `// Algorithm 21`

$\mathbf{V}_{hor}^{ortho} \leftarrow \emptyset$

for $\mathbf{v}^{(h)} \in \mathbf{V}_{hor}$ **do**

  $\mathbf{v}_u^{(h)} \leftarrow$ `image_to_gaussian_sphere`$(\mathbf{v}^{(h)}, f, \mathbf{p})$

  if $|\mathbf{v}_u^{(v)} \cdot \mathbf{v}_u^{(h)}| < \gamma_R$ **then**

    Append $\mathbf{v}^{(h)}$ to $\mathbf{V}_{hor}^{ortho}$

$\mathbf{V}_{hor}^{finite} \leftarrow \emptyset$

for $\mathbf{v}^{(h)} \in \mathbf{V}_{hor}$ **do**

  if $||\mathbf{v}^{(h)} - \mathbf{p}|| < \lambda$ **then**

    Append $\mathbf{v}^{(h)}$ to $\mathbf{V}_{hor}^{finite}$

if $\mathbf{V}_{hor}^{finite} == \emptyset$ **then**

  $\mathbf{V}_{hor}^{finite} \leftarrow \underset{\mathbf{v}^{(h)} \in \mathbf{V}_{hor}}{\arg\min}\left(||\mathbf{v}^{(h)} - \mathbf{p}||\right)$

if $\mathbf{V}_{hor}^{ortho} \bigcap \mathbf{V}_{hor}^{finite} \neq \emptyset$ **then**

  $\mathbf{V} \leftarrow \mathbf{V}_{hor}^{ortho} \bigcap \mathbf{V}_{hor}^{finite}$

else

  $\mathbf{V} \leftarrow \underset{\mathbf{v} \in \mathbf{V}_{hor}}{\arg\min}\big(\mathrm{NFA}(\mathbf{v})\big)$

Append $\mathbf{v}^{(v)}$ to $\mathbf{V}$

$\mathbf{l}_{hor} \leftarrow$ `horizon_line_voting`$(\mathbf{V})$ `// Algorithm 25`

Algorithm 24: Obtain vertical vanishing point candidates.

input : A list of candidate VPs in image domain $\mathbf{V}$. An angular threshold ω. The principal point $\mathbf{p}$.
output: A list $\mathbf{V}_{ver}$ of candidate vertical VPs.

$\mathbf{V}_{ver} \leftarrow \emptyset$
$\mathbf{l}_{ver} \leftarrow$ vertical line passing trough $\mathbf{p}$
for $\mathbf{v} \in \mathbf{V}$ **do**
    $\mathbf{v}_{centered} \leftarrow \mathbf{v} - \mathbf{p}$
    $\mathbf{l}_{\mathbf{v}} \leftarrow \overline{\mathbf{p}\mathbf{v}_{centered}}$
    if $\mathit{angle}(\mathbf{l}_{\mathbf{v}}, \mathbf{l}_{ver}) < \omega$ *AND* $|y_{\mathbf{v}}| > H$ **then**
        Append $\mathbf{v}$ to $\mathbf{V}_{ver}$

Algorithm 25: `horizon_line_voting`: Obtain horizon line by votes.

input : A list of horizontal VPs $\mathbf{V}_{hor}$, a vertical VP $\mathbf{v}_{ver}$ and their associated NFAs. The image center $\mathbf{p}$.
output: A line $\mathbf{l}_{hor}$ representing the horizon

$\mathbf{l}_{ver} \leftarrow \overline{\mathbf{p}\mathbf{v}_{ver}}$
$h_{hor} \leftarrow 0$
for $\mathbf{v}^{(i)} \in \mathbf{V}_{hor}$ **do**
    $w^{(i)} = \frac{-log_{10}(\mathrm{NFA}_i)}{\sum_j -log_{10}(\mathrm{NFA}_j)}$
    $\mathbf{l}_{hor}^{(i)} =$ line through $\mathbf{v}^{(i)}$ perpendicular to $\mathbf{l}_{ver}$
    $h_{hor} \leftarrow h_{hor} + w^{(i)} \cdot \texttt{height}(\mathbf{l}_{hor}^{(i)})$
$\mathbf{l}_{hor} \leftarrow$ line perpendicular to $\mathbf{l}_{ver}$ with height h_{hor}

Name	Description	Algorithm	Value
τ	Line segments length threshold	11	$\frac{\sqrt{W+H}}{1.7}$
θ	VP to line segment angle threshold	15, 16	1°
ζ	Variation threshold to detect ill-conditioned problem	15, 16	0.2
δ	Distance threshold to identify redundant VP detections	15, 17	0.02
f	Focal length in pixel units	18, 21, 22, 23	$\max(W, H)$
$\mathbf{p}$	Principal point or image center	18, 23, 25	$(H/2, W/2)$
γ_S	Strict angle threshold to determine orthogonality	18, 19, 20	85°
γ_R	Relaxed angle threshold to determine non-orthogonality	23	75°
ω	Angle threshold in image space to determine vertical VPs	23, 24	75°
λ	Distance threshold to determine infinite VPs	23	$5 \cdot W$

Table B.1: Parameters of the proposed method, with suggested values for general images. W and H are the image width and height.

B.8 Analysis of the method's parameters

There are ten parameters in the VP detection method. Table B.1 provides the description and values for general images, found empirically. Table B.2 presents the values optimized by grid search for York Urban Dataset and Eurasian Cities Dataset as explained in Section 4.3.2. The alignment detector is always used as described in Chapter 2, without modification or adjustment.

The most sensible parameter of the method is the focal length f. This value can sometimes be obtained if the camera model is known, or from the information included in the EXIF[1] header for some type of image formats. The EXIF header can provide the focal length in mm, the CCD size in mm and the CCD resolution in pixels, which can be used to compute the focal length in pixel units. If the camera is available but its characteristics unknown, f can be estimated by camera calibration Hartley and Zisserman [2003]. Otherwise, a general rule of thumb is to choose $f \in [0.3W, 3W]$, supposing W is the largest size of the image.

Also needed to compute the VP in the Gaussian sphere, the principal point of the camera, $\mathbf{p}$, can be safely assumed to be at the image center $[W/2, H/2]$.

The threshold τ for the length of line segments can have a big impact in the final result, if there happens to be a significant group of structures in the image whose length is close to the threshold. This threshold should vary with the image size. The chosen value of $\frac{\sqrt{W+H}}{1.7}$ is not strictly scale invariant, but, as most of the parameters in Table B.1, it has been found empirically using the benchmarking datasets of Section 4.3.2.

The refinement angle threshold θ is set to a very small angle, so that the segments chosen for refining a candidate VP do not largely differ from the points forming the alignment found by the alignment detector. This threshold can have a sensible impact in the final location of the refined VP. The threshold ζ is used to determine when the refinement has gone too far, although its sensibility is limited, since ill conditioned cases are a minority.

The angular threshold γ_S is used to determine whether a pair of unit vectors are close to orthogonal, in the case where orthogonality is searched for. On the other hand, γ_R is a more relaxed threshold, used to determine that two vectors are far from being orthogonal. The sensibility of these two parameters is in some way related to f. When f is known or correctly estimated, the mapping of vectors into the Gaussian sphere should be correct, so orthogonality and non-

[1] `http://en.wikipedia.org/wiki/Exchangeable_image_file_format`

Name	Algorithm	Value for YUD	value for ECD
τ	11	$\frac{\sqrt{W+H}}{1.7}$	$\frac{\sqrt{W+H}}{1.7}$
θ	15, 16	1.6°	0.375°
ζ	15, 16	0.15	0.3
δ	15, 17	0.004	0.02
f	18, 21, 22, 23	$1.05W$	$1.05W$
$\mathbf{p}$	18, 23, 25	$(307, 251)$	$(W/2, H/2)$
γ_S	18, 19, 20	0.06	N/A
γ_R	23	N/A	0.23
ω	23, 24	N/A	1.26°
λ	23	N/A	$3.6W$

Table B.2: Parameters of the proposed method trained for YUD and ECD datasets. H and W are the height and width of the image. A brief description of the parameters can be found in Table B.1.

orthogonality between vectors should be clearly determined. The values presented in table B.1 have been found empirically using the benchmarking datasets of Section 4.3.2.

The angular parameter ω is used to determine if a candidate VP could possibly be the vertical VP of the image. The idea is that a vertical VP should be in a close angle with respect to the vertical direction. Of course, in order for this trivial heuristic to work properly the image has to be approximately horizontal.

Finally, λ is a simple distance threshold used to determine if a point lies at "infinity". Naturally, this value needs to be proportional to the image size. This parameter can be sensible in the case where all the cues for horizontal VPs in the image come from parallel structures (the intersection of the lines lies at infinity). For example, when a building facade is photographed very closely, and no perspective is produced.

Bibliography

Ahuja, N. and Tuceryan, M. Extraction of early perceptual structure in dot patterns: Integrating region, boundary, and component Gestalt. *Computer Vision, Graphics, and Image Processing*, 48: 304–356, 1989.

Al-Maadeed, S., Ayouby, W., Hassaine, A., Almejali, A., Al-Yazeedi, A., and Al-Atiya, R. Arabic signature verification dataset. In *Proceedings of the International Arab Conference on Information Technology*, 2012.

Albert, M. K. and Hoffman, D. D. Genericity in spatial vision. In Luce, D., Romney, K., Hoffman, D., and D'Zmura M., editors, *Geometric Representations of Perceptual Phenomena: Arts. in Honor of Tarow Indow's 70th Birthday*, pages 95–112. Erlbaum, 1995.

Almansa, A., Desolneux, A., and Vamech, S. Vanishing point detection without any a priori information. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 25(4):502–507, 2003.

Amorese, D., Lagarde, J. L., and Laville, E. A point pattern analysis of the distribution of earthquakes in Normandy (France). *Bulletin of the Seismological Soc. of America*, 89(3):742–749, 1999.

Antunes, M. and Barreto, J. P. A global approach for the detection of vanishing points and mutually orthogonal vanishing directions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. doi: 10.1109/CVPR.2013.176.

Arcasoya, A., Toprakb, V., and Kaymakçı, N. Comprehensive strip based lineament detection method (COSBALID) from point-like features: a GIS approach. *Computers & Geosciences*, 30: 45–57, 2004.

Arp, H. and Hazard, C. Peculiar configurations of quasars in two adjacent areas of the sky. *The Astrophysical Journal*, 240(3):726–736, 1980.

Ballard, D. H. and Brown, C. M. *Computer Vision*. Prentice-Hall, 1982.

Barinova, O., Lempitsky, V., Tretiak, E., and Kohli, P. Geometric image parsing in man-made environments. In *European Conference on Computer Vision (ECCV)*. Springer Berlin Heidelberg, 2010.

Barnard, S. T. Interpreting perspective images. *Artificial intelligence*, 21(4):435–462, 1983.

Bazin, J.-C., Seo, Y., Demonceaux, C., Vasseur, P., Ikeuchi, K., Kweon, I., and Pollefeys, M. Globally optimal line clustering and vanishing point estimation in Manhattan world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. doi: 10.1109/CVPR.2012.6247731.

Black, P. E. "bucket". Dictionary of Algorithms and Data Structures [http://www.nist.gov/dads/HTML/bucket.html], Vreda Pieterse and Paul E. Black, eds. 8 January 2004 (accessed March 18, 2015).

Broadbent, S. Simulating the Ley Hunter. *Journal of the Royal Statistical Soc. Series A*, 143(2): 109–140, 1980.

Caelli, T. M., Preston, G., and Howell, E. Implications of spatial summation models for processes of contour perception: A geometric perspective. *Vision Research*, 18(6):723–734, 1978.

Canny, J. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 8(6):679–698, 1986.

Cao, F. Application of the Gestalt principles to the detection of good continuations and corners in image level lines. *Computing and Visualization in Science*, 7:3–13, 2004.

Cao, F., Delon, J., Desolneux, A., Musé, P., and Sur, F. A unified framework for detecting groups and application to shape recognition. *Journal of Mathematical Imaging and Vision*, 27(2): 91–119, 2007.

Carpenter, G. A. and Grossberg, S. A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer vision, graphics, and image processing*, 37(1):54–115, 1987.

Chui, H. and Rangarajan, A. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 89(2):114–141, 2003.

Collins, R. T. and Weiss, R. S. Vanishing point calculation as a statistical inference on the unit sphere. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 1990.

Comaniciu, D. and Meer, P. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analyisis and Machine Intelligence (TPAMI)*, 24(5):603–619, 2002.

Compton, B. J. and Logan, G. D. Evaluating a computational model of perceptual grouping by proximity. *Perception & Psychophysics*, 53(4):403–421, 1993.

Coughlan, J. M. and Yuille, A. L. Manhattan world: Orientation and outlier detection by Bayesian inference. *Neural Computation*, 15(5):1063–1088, 2003.

Crow, F. C. Summed-area tables for texture mapping. In *ACM SIGGRAPH Computer Graphics*, volume 18, pages 207–212. ACM, 1984.

Danuser, G. and Stricker, M. Parametric model fitting: From inlier characterization to outlier detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(2):263–280, 1998.

Davé, R. N. Use of the adaptive fuzzy clustering algorithm to detect lines in digital images. *Intelligent Robots and Computer Vision VIII*, 1192:600–611, 1989.

Demeyer, M. and Machilsen, B. The construction of perceptual grouping displays using gert. *Behavior research methods*, 44(2):439–446, 2012.

Denis, P., Elder, J. H., and Estrada, F. J. Efficient edge-based methods for estimating Manhattan frames in urban imagery. In *European Conference on Computer Vision (ECCV)*, 2008.

Desolneux, A., Moisan, L., and Morel, J.-M. Meaningful alignments. *International Journal of Computer Vision*, 40(1):7–23, 2000.

Desolneux, A., Moisan, L., and Morel, J. A grouping principle and four applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003.

Desolneux, A., Moisan, L., and Morel, J. *From Gestalt Theory to Image Analysis, a Probabilistic Approach*. Springer, 2008.

Dubská, M., Herout, A., and Havel, J. PClines - line detection using parallel coordinates. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

Dubská, M., Herout, A., and Havel, J. Real-time precise detection of regular grids and matrix codes. *Journal of Real-Time Image Processing*, 2013. doi: 10.1007/s11554-013-0325-6.

Dubuc, B. and Zucker, S. W. Complexity, confusion, and perceptual grouping. part i: The curve-like representation. *Journal of Mathematical Imaging and Vision*, 15(1-2):55–82, 2001a.

Dubuc, B. and Zucker, S. W. Complexity, confusion, and perceptual grouping. part ii: Mapping complexity. *International Journal of Computer Vision*, 42(1-2):83–115, 2001b.

Duda, R. O. and Hart, P. E. Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1972.

Dy, J. G. and Brodley, C. E. Feature selection for unsupervised learning. *Journal of Machine Learning Research (JMLR)*, 5:845–889, 2004.

Edmunds, M. G. and H., G. G. Random alignment of quasars. *Nature*, 290:481–483, 1981.

Elder, J. and Zucker, S. The effect of contour closure on the rapid discrimination of two-dimensional shapes. *Vision research*, 33(7):981–991, 1993.

Elder, J. and Zucker, S. A measure of closure. *Vision research*, 34(24):3361–3369, 1994.

Elder, J. H. and Zucker, S. W. Computing contour closure. In *Computer Vision—ECCV'96*, pages 399–412. Springer, 1996.

Elder, J. H. and Zucker, S. W. Evidence for boundary-specific grouping. *Vision Research*, 38(1): 143–152, 1998.

Elhamifar, E. and Vidal, R. Sparse subspace clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2790–2797, 2009.

Ellis, W., editor. *A Source Book of Gestalt Psychology*. Humanities Press, 1967 (originally 1938).

Escalera, A. d. l. and Armingol, J. M. Automatic chessboard detection for intrinsic and extrinsic camera parameter calibration. *Sensors*, 10:2027–2044, 2010.

Feldman, J. Regularity-based perceptual grouping. *Computational Intelligence*, 13(4):582–623, 1997a.

Feldman, J. Bayesian contour integration. *Attention, Perception, & Psychophysics*, 63:1171–1182, 2001. ISSN 1943-3921.

Feldman, J. Curvilinearity, covariance, and regularity in perceptual groups. *Vision Research*, 37 (20):2835–2848, 1997b.

Feldman, J. Formation of visual "objects" in the early computation of spatial relations. *Perception & Psychophysics*, 69(5):816–827, 2007.

Field, D. J., Hayes, A., and Hess, R. F. Contour integration by the human visual system: Evidence for a local "association field". *Vision research*, 33(2):173–193, 1993.

Figueiredo, M. A. T. and Jain, A. K. Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):381–396, 2002.

Fischler, M. A. and Bolles, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24 (6):381–395, 1981.

French, R. S. Pattern recognition in the presence of visual noise. *Journal of experimental psychology*, 47(1):27, 1954.

Frigui, H. and Krishnapuram, R. A robust competitive clustering algorithm with applications in computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):450–465, 1999.

Gerig, G. Linking image-space and accumulator-space: A new approach for object-recognition. *First International Conference on Computer Vision ICCV*, pages 112–117, 1987.

Gerig, G. and Klein, F. Fast contour identification through efficient hough transform and simplified interpretation strategy. In *Proc. Int. Conf: Pattern Rec.*, volume 1, pages 498–500, 1986.

Gigus, Z. and Malik, J. *Detecting curvilinear structure in images*. University of California, Berkeley, Computer Science Division, 1991.

Glass, L. et al. Moiré effect from random dots. *Nature*, 223(5206):578–580, 1969.

Goh, A. and Vidal, R. Segmenting motions of different types by unsupervised manifold clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–6, 2007.

Gong, D. and Medioni, G. Probabilistic tensor voting for robust perceptual grouping. In *IEEE Computer Vision and Pattern Recognition Workshops (CVPRW)*,, pages 1–8, 2012.

Gong, D., Zhao, X., and Medioni, G. Robust multiple manifolds structure learning. In *International Conference on Machine Learning (ICML)*, pages 321–328, 2012.

Gori, S. and Spillmann, L. Detection vs. grouping thresholds for elements differing in spacing, size and luminance. an alternative approach towards the psychophysics of gestalten. *Vision research*, 50(12):1194–1202, 2010.

Grompone von Gioi, R. and Jakubowicz, J. On computational Gestalt detection thresholds. *Journal of Physiology – Paris*, 103(1–2):4–17, 2009.

Grompone von Gioi, R., Jakubowicz, J., Morel, J. M., and Randall, G. LSD: a line segment detector. *Image Processing On Line*, 2012. doi: 10.5201/ipol.2012.gjmr-lsd.

Grossberg, S. and Mingolla, E. Neural dynamics of perceptual grouping: textures, boundaries, and emergent segmentations. *Advances in Psychology*, 43:144–210, 1987.

Guy, G. and Medioni, G. Perceptual grouping using global saliency-enhancing operators. In *International Conference on Pattern Recognition (ICPR)*, pages I:99–103, 1992.

Guy, G. and Medioni, G. Inferring global perceptual contours from local features. In *Computer Vision and Pattern Recognition, 1993. Proceedings CVPR'93., 1993 IEEE Computer Society Conference on*, pages 786–787. IEEE, 1993.

Hall, P., Tajvidi, N., and E., M. P. Locating lines among scattered points. *Bernoulli*, 12(5): 821–839, 2006.

Hammer, O. New statistical methods for detecting point alignments. *Computers & Geosciences*, 35:659–666, 2009.

Hammer, O., Harper, D. A. T., and Ryan, P. D. PAST: Paleontological statistics software package for education and data analysis. *Palaeontologia Electronica*, 4(1), 2001.

Harris, C. and Stephens, M. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Manchester, UK, 1988.

Hartley, R. and Zisserman, A. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

Havel, J., Herout, A., and Dubská, M. Vanishing points in point-to-line mappings and other line parameterizations. *Pattern Recognition Letters*, 34:703–708, 2013.

Herault, L. and Horaud, R. Figure-ground discrimination: A combinatorial optimization approach. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(9):899–914, 1993.

Hoeffding, W. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. doi: 10.1080/01621459.1963.10500830.

Hough, P. V. C. Machine analysis of bubble chamber pictures. *International Conference on High Energy Accelerators and Instrumentation*, 1:554–556, 1959.

Hough, P. V. C. *Method and Means for Recognizing Complex Patterns*. U.S. Patent 3,069,654, Dec. 18, 1962.

Jain, A. K. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 2010.

Kanizsa, G. *Organization in vision: Essays on Gestalt perception*. Praeger New York:, 1979.

Kanizsa, G. *Grammatica del vedere: saggi su percezione e gestalt*. Il Mulino, 1980.

Kanizsa, G. *Vedere e pensare*. Il Mulino, 1991.

Kendall, D. G. and Kendall, W. S. Alignments in two-dimensional random sets of points. *Advances in Applied probability*, 12:380–424, 1980.

Kersten, D., Mamassian, P., and Yuille, A. Object perception as bayesian inference. *Annual Review of Psychology*, 55(1):271–304, 2004. ISSN 0066-4308.

Kiryati, N. and M., B. A. On navigating between friends and foes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):602–606, 1991.

Kiryati, N. and M., B. A. What's in a set of points? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(4):496–500, 1992.

Kiryati, N., Eldar, Y., and Bruckstein, A. M. A probabilistic hough transform. *Pattern recognition*, 24(4):303–316, 1991.

Kiryati, N., Henricsson, O., and Rosenthaler, L. On the perception of dotted lines. Technical Report BIWI-TR-137, Inst. for Communications Technology, Image Science Laboratory, 1992.

Kiryati, N., Kälviäinen, H., and Alaoutinen, S. Randomized or probabilistic hough transform: unified performance evaluation. *Pattern Recognition Letters*, 21(13):1157–1164, 2000.

Köhler, W. *Gestalt Psychology*. Liveright, 1947.

Kosecká, J. and Zhang, W. Video compass. In *European Conference on Computer Vision (ECCV)*, 2002.

Kubovy, M., Holcombe, A. O., and Wagemans, J. On the lawfulness of grouping by proximity. *Cognitive psychology*, 35(1):71–98, 1998.

Lezama, J., Grompone von Gioi, R., Randall, G., and Morel, J. M. Finding vanishing points via point alignments in image primal and dual domains. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 509–515, 2014a.

Lezama, J., Grompone von Gioi, R., Randall, G., and Morel, J. A contrario detection of good continuation of points. In *IEEE International Conference on Image Processing (ICIP) 2014*. IEEE, 2014b.

Lezama, J., Randall, G., Morel, J., and Grompone von Gioi, R. Good continuation in dot patterns: a quantitative approach based on local symmetry and non-accidentalness. *Vision Research*, submitted, 2015a.

Lezama, J., Blusseau, S., Morel, J.-M., Randall, G., and von Gioi, R. G. Psychophysics, gestalts and games. In *Neuromathematics of Vision*, pages 217–242. Springer, 2014c.

Lezama, J., Morel, J.-M., Randall, G., and Grompone von Gioi, R. A Contrario 2D Point Alignment Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2014d. doi: 10.1109/TPAMI.2014.2345389. Published online on August 5, 2014. http://dx.doi.org/10.1109/TPAMI.2014.2345389.

Lezama, J., Randall, G., Morel, J.-M., and Grompone von Gioi, R. An Unsupervised Point Alignment Detection Algorithm. *Image Processing On Line*, submitted, 2014e.

Lezama, J., Randall, G., Morel, J.-M., and Grompone von Gioi, R. An Algorithm for Detecting Vanishing Points via Point Alignments. *Image Processing On Line*, submitted, 2015b.

Litton, C. D. and Restorick, J. Computer analysis of post hole distributions. *Computer Applications in Archaeology*, pages 85–92, 1983.

Logan, G. D. The code theory of visual attention: an integration of space-based and object-based attention. *Psychological review*, 103(4):603, 1996.

Loss, L., Bebis, G., Nicolescu, M., and Skourikhine, A. Perceptual grouping based on iterative multi-scale tensor voting. In *Advances in Visual Computing*, pages 870–881. Springer, 2006.

Loss, L., Bebis, G., Nicolescu, M., and Skurikhin, A. An iterative multi-scale tensor voting scheme for perceptual grouping of natural shapes in cluttered backgrounds. *Computer Vision and Image Understanding (CVIU)*, 113(1):126 – 149, 2009. ISSN 1077-3142. doi: http://dx.doi.org/10.1016/j.cviu.2008.07.011.

Lowe, D. *Perceptual Organization and Visual Recognition*. Kluwer Academic Publishers, 1985.

Lowe, D. and Binford, T. O. Segmentation and aggregation: an approach to figure-ground phenomena. *Image Partitioning and Perceptual Organization*, pages 282–292, 1982.

Lutz, T. M. An analysis of the orientations of large-scale crustal structures: A statistical approach based on areal distributions of pointlike features. *Journal of Geophysical Research*, 91(B1):421–434, 1986.

Machilsen, B. and Wagemans, J. Integration of contour and surface information in shape detection. *Vision research*, 51(1):179–186, 2011.

Mack, C. The expected number of aggregates in a random distribution of n points. *Mathematical Proceedings of the Cambridge Philosophical Soc.*, 46(2):285–292, 1950.

Marr, D. *Vision*. Freeman and co., 1982.

Metzger, W. *Gesetze des Sehens*. Verlag Waldemar Kramer, Frankfurt am Main, third edition, 1975.

Metzger, W. *Laws of Seeing*. The MIT Press, 2006 (originally 1936). English translation of the first edition of Metzger [1975].

Miles, R. E. On the homogeneous planar Poisson point process. *Mathematical Biosciences*, 6: 85–127, 1970.

Mirzaei, F. M. and Roumeliotis, S. I. Optimal estimation of vanishing points in a Manhattan world. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011.

Mordohai, P. and Medioni, G. Tensor voting: a perceptual organization approach to computer vision and machine learning. *Synthesis Lectures on Image, Video, and Multimedia Processing*, 2(1): 1–136, 2006.

Mumford, D. Pattern theory: the mathematics of perception. *Proceedings of the International Congress of Mathematicians, Beijing*, I:401–422, 2002.

Murtagh, F. and Raftery, A. E. Fitting straight lines to point patterns. *Pattern Recognition*, 17 (5):479–483, 1984.

Musé, P., Sur, F., Cao, F., Gousseau, Y., and Morel, J.-M. An a contrario decision method for shape element recognition. *International Journal of Computer Vision*, 69(3):295–315, 2006.

Mussap, A. J. and Levi, D. M. Amblyopic deficits in detecting a dotted line in noise. *Vision Research*, 40:3297–3307, 2000.

Myaskouvskey, A., Gousseau, Y., and Lindenbaum, M. Beyond independence: An extension of the a contrario decision procedure. *International journal of computer vision*, 101(1):22–44, 2013.

Ng, A. Y., Jordan, M. I., Weiss, Y., et al. On spectral clustering: Analysis and an algorithm. *Neural Information Processing Systems (NIPS)*, 2:849–856, 2002.

Palmer, S. E. *Vision Science: Photons to Phenomenology*. The MIT Press, 1999.

Parent, P. and Zucker, S. W. Trace inference, curvature consistency, and curve detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(8):823–839, 1989.

Perona, P. and Freeman, W. A factorization approach to grouping. In *Computer Vision—ECCV'98*, pages 655–670. Springer, 1998.

Pinar Saygin, A., Cicekli, I., and Akman, V. Turing test: 50 years later. *Minds and Machines*, 10 (4):463–518, 2000.

Pizlo, Z., Salach-Golyska, M., and Rosenfeld, A. Curve detection in a noisy image. *Vision Research*, 37(9):1217–1241, 1997.

Preiss, A. K. *A Theoretical and Computational Investigation into Aspects of Human Visual Perception: Proximity and Transformations in Pattern Detection and Discrimination*. PhD thesis, University of Adelaide, 2006.

Prinzmetal, W. and Banks, W. P. Good continuation affects visual detection. *Perception & Psychophysics*, 21(5):389–395, 1977.

Quan, L. and Mohr, R. Determining perspective structures using hierarchical Hough transform. *Pattern Recognition Letters*, 9:279–286, 1989.

Rother, C. A new approach to vanishing point detection in architectural environments. *Image and Vision Computing*, 20(9):647–655, 2002.

Sarkar, S. and Boyer, K. L. Perceptual organization in computer vision: A review and a proposal for a classificatory structure. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(2):382–399, 1993.

Schindler, G. and Dellaert, F. Atlanta world: An expectation maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

Sha'asua, A. and Ullman, S. Structural saliency: The detection of globally salient structures using a locally connected network. In *Computer Vision., Second International Conference on*, pages 321–327. IEEE, 1988.

Small, C. G. Techniques of shape analysis on sets of points. *International Statistical Review*, 56 (3):243–257, 1988.

Smits, J., Vos, P. G., and Van Oeffelen, M. P. The perception of a dotted line in noise: a model of good continuation and some experimental results. *Spatial Vision*, 1(2):163–177, 1984.

Smits, J. and Vos, P. A model for the perception of curves in dot figures: the role of local salience of "virtual lines". *Biological cybernetics*, 54(6):407–416, 1986.

Souvenir, R. and Pless, R. Manifold clustering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, volume 1, pages 648–653. IEEE, 2005.

Tardif, J.-P. Non-iterative approach for fast and accurate vanishing point detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2009.

Tepper, M. and Sapiro, G. A biclustering framework for consensus problems. *SIAM Journal on Imaging Sciences*, 7(4):2488–2525, 2014a. doi: 10.1137/140967325.

Tepper, M. and Sapiro, G. Intersecting 2d lines: A simple method for detecting vanishing points. In *IEEE International Conference on Image Processing (ICIP)*, 2014b.

Tepper, M., Musé, P., and Almansa, A. Meaningful clustered forest: an automatic and robust clustering algorithm. *arXiv preprint arXiv:1104.0651*, 2011.

Thrift, P. R. and Dunn, S. M. Approximating point-set images by line segments using a variation of the Hough transform. *Computer Vision, Graphics, and Image Processing*, 21:383–394, 1983.

Tripathy, S. P., Mussap, A. J., and Barlow, H. B. Detecting collinear dots in noise. *Vision Research*, 39:4161–4171, 1999.

Turing, A. Computing machinery and intelligence. *Mind*, 59:433–460, 1950.

Uttal, W. R. The effect of deviations from linearity on the detection of dotted line patterns. *Vision Res*, 13(11):2155–63, 1973. ISSN 0042-6989.

Uttal, W. R. *An Autocorrelation Theory of Form Detection*. John Wiley & Sons, 1975.

Uttal, W. R. *The Perception of Dotted Forms*. Erlbaum, 1987.

Uttal, W., Bunnell, L., and Corwin, S. On the detectability of straight lines in visual noise: An extension of french's paradigm into the millisecond domain. *Perception and Psychophysics*, 8: 385–388, 1970. ISSN 0031-5117.

van Assen, M. A. and Vos, P. G. Evidence for curvilinear interpolation from dot alignment judgements. *Vision research*, 39(26):4378–4392, 1999.

van Oeffelen, M. P. and Vos, P. G. An algorithm for pattern description on the level of relative proximity. *Pattern Recognition*, 16(3):341–348, 1983.

Vanegas, M. C., Bloch, I., and Inglada, J. Detection of aligned objects for high resolution image understanding. pages 464–467, 2010.

Vedaldi, A. and Zisserman, A. Self-similar sketch. In *European Conference on Computer Vision (ECCV)*, 2012.

Viola, P. and Jones, M. Rapid object detection using a boosted cascade of simple features. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1:I–511, 2001.

Von Luxburg, U. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

Wadge, G. and Cross, A. Quantitative methods for detecting aligned points: An application to the volcanic vents of the Michoacan-Guanajuato volcanic field, Mexico. *Geology*, 16:815–818, 1988.

Wagemans, J. Perceptual use of nonaccidental properties. *Canadian Journal of Psychology*, 46(2): 236–279, 1992.

Wagemans, J., Elder, J. H., Kubovy, M., Palmer, S. E., Peterson, M. A., Singh, M., and von der Heydt, R. A century of gestalt psychology in visual perception: I. perceptual grouping and figure–ground organization. *Psychological Bulletin*, 138(6):1172–1217, 2012a.

Wagemans, J., Feldman, J., Gepshtein, S., Kimchi, R., Pomerantz, J. R., van der Helm, P. A., and van Leeuwen, C. A century of gestalt psychology in visual perception: II. conceptual and theoretical foundations. *Psychological Bulletin*, 138(6):1218–1252, 2012b.

Wang, Y., Jiang, Y., Wu, Y., and Zhou, Z.-H. Spectral clustering on multiple manifolds. *Neural Networks, IEEE Transactions on*, 22(7):1149–1161, 2011.

Wertheimer, M. Untersuchungen zur Lehre von der Gestalt. II. *Psychologische Forschung*, 4(1): 301–350, 1923. An abridged translation to English is included in Ellis [1967 (originally 1938)].

Wildenauer, H. and Hanbury, A. Robust camera self-calibration from monocular images of Manhattan worlds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. doi: 10.1109/CVPR.2012.6248008.

Williams, L. R. and Thornber, K. K. A comparison of measures for detecting natural shapes in cluttered backgrounds. *International Journal of Computer Vision (IJCV)*, 34(2-3):81–96, 1999.

Witkin, A. P. and Tenenbaum, J. M. What is perceptual organization for? *International Joint Conference on Artificial Intelligence (IJCAI)*, 2:1023–1026, 1983a.

Witkin, A. P. and Tenenbaum, J. M. On the role of structure in vision. In Beck, J., Hope, B., and Rosenfeld, A., editors, *Human and Machine Vision*, pages 481–543. Academic Press, 1983b.

Xia, G.-S., Delon, J., and Gousseau, Y. Accurate junction detection and characterization in natural images. *International Journal of Computer Vision*, 106(1):31–56, 2014.

Xu, Y., Oh, S., and Hoogs, A. A minimum error vanishing point detection approach for uncalibrated monocular images of man-made environments. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

Yan, J. and Pollefeys, M. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *European Conference on Computer Vision (ECCV)*, pages 94–106. Springer, 2006.

Zhang, D. and Lutz, T. Structural control of igneous complexes and kimberlites: a new statistical method. *Tectonophysics*, 159:137–148, 1989.

Zhao, Y.-G., Wang, X., Feng, L.-B., Chen, G., Wu, T.-P., and Tang, C.-K. Calculating vanishing points in dual space. In *Intelligent Science and Intelligent Data Engineering*, volume 7751 of *Lecture Notes in Computer Science*, pages 522–530. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-36668-0. doi: 10.1007/978-3-642-36669-7_64.

Zuiderwijk, E. J. Alignment of randomly distributed objects. *Nature*, 295:577–578, 1982.