Exploitation d'indices visuels liés au mouvement pour l'interprétation du contenu des séquences vidéos

par

Mathieu Marquis Bolduc

mémoire présenté au Département d'informatique
en vue de l'obtention du grade de maître ès sciences (M.Sc.)

FACULTÉ DES SCIENCES
UNIVERSITÉ DE SHERBROOKE

Sherbrooke, Québec, Canada, mai 2008

La page 64 est blanche.
Complet tel quel.

Your file Votre référence
ISBN: 978-0-494-49540-7
Our file Notre référence
ISBN: 978-0-494-49540-7

Le 8 mai 2008

le jury a accepté le mémoire de M. Mathieu Marquis-Bolduc dans sa version finale.

Membres du jury

M. François Deschênes
Directeur
Département d'informatique

M. Madjid Allili
Membre
- Bishop's University

M. Pierre-Marc Jodoin
Président-rapporteur
Département d'informatique

SOMMAIRE

L'interprétation du contenu des séquences vidéo est un des principaux domaines de recherche en vision artificielle. Dans le but d'enrichir l'information provenant des indices visuels qui sont propres à une seule image, on peut se servir d'indices découlant du mouvement entre les images. Ce mouvement peut être causé par un changement d'orientation ou de position du système d'acquisition, par un déplacement des objets dans la scène, et par bien d'autres facteurs. Je me suis intéressé à deux phénomènes découlant du mouvement dans les séquences vidéo. Premièrement, le mouvement causé par la caméra, et comment il est possible de l'interpréter par une combinaison du mouvement apparent entre les images, et du déplacement de points de fuite dans ces images. Puis, je me suis intéressé à la détection et la classification du phénomène d'occultation, qui est causé par le mouvement dans une scène complexe, grâce à un modèle géométrique dans le volume spatio-temporel. Ces deux travaux sont présentés par le biais de deux articles soumis pour publication dans des revues scientifiques.

Remerciements

Je tiens à remercier tout d'abord mon directeur de maîtrise, le professeur François Deschênes, pour sa dévotion à ses étudiants, ses encouragements et son expertise. Je tiens également à remercier les professeurs Djemel Ziou et Marie-Flavie Auclair-Fortier pour leur aide et leurs conseils, ainsi que ma famille pour leur soutien. Finalement, il me faut remercier mes collègues Wei Pan, Ian Bailey, Charles Perrault et Daniel Lévesque. Votre apport, votre aide et votre soutien furent grandement appréciés.

TABLE DES MATIÈRES

Page

SOMMAIRE ii

REMERCIEMENTS iii

TABLE DES MATIÈRES iv

LISTE DES TABLEAUX vi

LISTE DES FIGURES vii

Introduction 1

Chapitre 1 — Mouvement et Perspective 4

1.1 Avant-Propos 4

1.2 Combining Apparent Motion and Perspective as Visual Cues for Content-based Camera Motion Indexing 7

Chapitre 2 — Occultations dans le volume spatio-temporel **44**

2.1 Avant-Propos 44

2.2 Occlusion Event Detection using Geometric Features in Spatio-temporal Volumes 47

Conclusion **65**

BIBLIOGRAPHIE **67**

LISTE DES TABLEAUX

		Page
I-1	Individual camera motion detection results for real video sequences	34
I-2	Combined camera motion detection results for real video sequences using the proposed method	35
I-3	Camera motion detection result for a trombone shot from the movie Vertigo	38
II-1	Occlusion events detection results for all video sequences, with occlusion velocity ranging from 0.3 to 18 pixels/frame	57
II-2	Occlusion events detection results for video sequences with relative occlusion velocity ranging between 1 and 5 pixels per frame	58
II-3	Occlusion events classification results for video sequences with relative event velocity between 0.5 and 18 pixels per frame	60

LISTE DES FIGURES

		Page
I-1	Box under different projections	14
I-2	Overview of the algorithm for camera motion extraction based on apparent motion and vanishing points	27
I-3	Frames from the three-motion artificial sequence at time t = 0 and 14	31
I-4	Correlation between the set of ground truth parameters (MGTP) and the set of estimated parameters (MEP) for all of the seven motions	32
I-5	Frames from the stairway sequence at time t = 0, 10 and 20	37
I-6	Example of optical flow obtained at t = 10 from the stairway sequence	37
II-1	An occlusion event occurs as one object moves in front of the other	48
II-2	An EPI image of a video sequence featuring several occlusion events	49
II-3	Two rectangles move toward each other	50
II-4	The principal curvature direction of two objects moving in the same direction but at different speed	51
II-5	Example of non-occlusion ridges caused by the motion of simple forms	53

		Page
II-6	Closeup view of an occlusion event as the hand pass over the white band	55
II-7	A test sequences with both objects are moving toward each other	56
II-8	A complex video sequence featuring dozens of occlusion events	57
II-9	The spatio-temporal volume from the sequence shown in Figure 7	59
II-10	A cross view of the classication of the collision event presented in Figure 7	59
II-11	A test sequence features a textured box moving over color stripes	60
II-12	Spatiotemporal surface orientation variability for the occlusion event presented in Figure 7	60
II-13	Effect of gaussian noise on occlusion events detection rate and false detection rate	60

Introduction

La vision artificielle est un domaine de recherche complexe qui s'intéresse à la compréhension par un système informatique du contenu d'une image ou d'une séquence d'images. L'importance des systèmes de vision artificielle dans notre société ne cesse d'augmenter. La vision artificielle comprend des applications très diversifiées comme la reconstruction de scènes, le suivi d'objet ou la reconnaissance de visages. Elle peut ainsi être exploitée dans des secteurs comme l'aérospatial, les transports, la santé, le contrôle de qualité et le divertissement. Plusieurs de ces applications dépendent de l'analyse du contenu 3D des images, i.e. les relations entre les divers points et objets dans la scène 3D représentée dans ces images. Pour ce faire, on peut se servir de différents indices visuels présents dans une image. Toutefois, lorsque le système de vision travaille avec une séquence vidéo, il est aussi possible de se servir du mouvement afin non seulement d'augmenter les indices visuels mais aussi d'extraire une toute nouvelle gamme d'informations sémantiques.

Dans cette optique, je me suis intéressé à l'importance du mouvement dans les séquences vidéo. Il est raisonnable de penser que la plupart des événements notables présents dans ces séquences se déroulent sur une période de temps excédant celle de la prise d'une seule image. Toutefois, la prise des séquences vidéo digitalisées passe par une discrétisation autant spatiale que temporelle qui présente des défis intéressants pour la détection et l'analyse de ces événements. Il existe plusieurs causes, dans n'importe quelle scène, qui

peuvent créer du mouvement. Certaines causes sont kynétiques, comme le mouvement de la caméra ou le déplacement du sujet. D'autres causes sont plus subtiles, comme le changement de certains paramètres caméra ou même un changement d'illumination. Je me suis intéressé tout d'abord au mouvement de la caméra qui tend a créer un effet de mouvement global dans la séquence vidéo. Cette information est d'un intérêt particulier dans le domaine de la cinématographie, car elle est fortement liée au jargon déjà utilisé pour décrire les séquences vidéo [3]. Par la suite, je me suis intéressé à un phénomène qui résulte à la fois du mouvement caméra et objet, et de la projection d'une scène en trois dimensions sur un plan image en deux dimensions. Il s'agit du phénomène d'occultation, lorsqu'un objet est occulté par un second objet. Comme nous le décrirons subséquemment, les occultations présentent à la fois un défi et un potentiel d'information sémantique important dans le contexte de l'analyse de séquences vidéo.

Concernant le mouvement caméra, mon apport se situe au niveau de la reconnaissance et de la catégorisation de ce mouvement. En cinématographie, il existe 7 principaux types de mouvement caméra : trois mouvements de translation, trois mouvements de rotation et un mouvement de zoom. Cependant, certains de ces mouvements présentent un effet similaire à l'écran, ce qui rend la reconnaissance de n'importe quels des sept mouvements difficile, surtout lorsqu'ils sont combinés [13]. Mon apport fut de combiner le mouvement apparent avec les informations de perspective, sous la forme de points de fuites. Ceux-ci sont présents dans quantités de scènes se déroulant dans un environnement humain formé d'angles droits, ce que certains appellent l'hypothèse du monde de Manhattan [6]. Les caractéristiques intrinsèques des points de fuites ainsi que leur déplacement dans la séquence vidéo, lorsque combinés avec le mouvement apparent, permettent à la méthode proposée de discerner les mouvements caméra énumérés ci-haut, incluant des combinaisons allant jusqu'à trois mouvements.

Concernant le phénomène d'occultation, mon intérêt se situe au niveau de la détection de

l'événement d'occultation: lorsqu'un objet disparaît ou apparaît derrière un autre objet, dû au mouvement, ainsi qu'à l'identification de contours d'occultations. Ces événements, conséquence inévitable du mouvement et d'une scène complexe, présentent un intérêt particulier. Premièrement, ils ont une valeur sémantique intrinsèque, pouvant être utilisée directement pour décrire une scène. Deuxièmement, ils peuvent être utilisés pour décrire les relations entre les différents objets dans la scène [22], ainsi que pour définir les limites physiques des objets. Finalement, ils peuvent influencer positivement ou négativement certaines applications de vision artificielle telles que le suivi d'objet [11]. Mon travail fut d'examiner la géométrie de ces événements dans le volume spatio-temporel afin de les détecter et de les discerner d'autres événements.

Dans un premier temps, nous présentons mes travaux sur la reconnaissance du mouvement de la caméra. Par la suite, nous présentons mon travail concernant la détection et la classification des événements d'occultations.

Chapitre 1

Mouvement et Perspective

1.1 Avant-Propos

Dans le but d'enrichir les méthodes d'interprétation de séquences vidéo à partir du contenu visuel, il est nécessaire de pouvoir identifier et extraire des informations visuelles dont la présence dans les images peut être associée aux relations 3D des objets présents dans la scène. En cinématographie, on reconnaît que la scène comporte généralement trois composantes fondamentales [3]. Premièrement, il y a le sujet. Ensuite, vient la distance ou cadrage du sujet. Notez que ce sont aussi des composantes qui se retrouvent en photographie, ce qui n'est pas le cas de la dernière composante: le déplacement de la caméra. C'est à l'exploitation de cette dernière que s'intéresse la méthode décrite dans l'article qui suit, intitulé «Combining Apparent Motion and Perspective as Visual Cues for Content-based Camera Motion Indexing», lequel fut accepté pour publication dans sa version finale en 2007 dans la revue *Pattern Recognition.*

Cet article présente une nouvelle méthode pour l'identification automatique des mouvements de la caméra dans une séquence vidéo. Il existe sept types de mouvement caméra.

Les trois premiers sont des mouvements de translation: le «Tracking», un mouvement de translation vers les côtés de la caméra, le «Booming», un mouvement de translation vers le haut ou le bas, et le «Dollying», une translation vers l'avant dans la direction de l'axe optique. Les trois mouvements suivants sont des mouvements de rotation : le «Panning», une rotation autour de l'axe vertical de la caméra, le «Tilting», une rotation de haut en bas (similaire à un hochement de tête), et le «Rolling», une rotation autour de l'axe optique de la caméra. Ces six premiers mouvements sont causés par un mouvement de l'ensemble de la caméra. Le dernier mouvement, le zoom, n'est pas créé par un déplacement de la caméra mais plutôt par le mouvement de composantes physiques de la caméra (e.g, la lentille). Toutefois, son effet à l'écran se compare à celui d'un mouvement extrinsèque, dans la perspective où son effet est global et important visuellement. Le mouvement de la caméra revêt une importance sémantique non-négligeable et, lorsque présent, sert à la description du contenu d'une séquence vidéo. Son identification passe généralement par l'analyse globale du mouvement entre chaque image consécutive (mouvement apparent) [7, 16, 20, 21]. Cependant, certaines paires de mouvement caméra peuvent causer un mouvement apparent très similaire: un «panning»peut fortement ressembler à un «tracking»lorsque l'ouverture de la caméra est faible. De la même manière, le mouvement apparent causé par un «booming»peut être extrêmement similaire à celui causé par un «tilting». En particulier, le zoom se révèle extrêmement difficile à différencier d'un «dollying»si nous ne considérons que le mouvement apparent (projection en 2D du mouvement 3D). À notre connaissance, aucune méthode d'identification des mouvements de la caméra n'est capable de les distinguer.

Ces constats nous ont amenés à combiner l'indice du mouvement apparent avec un autre indice: la perspective [5]. La géométrie perpendiculaire caractéristique des environnements humains fait en sorte que les points de fuites y sont facilement détectables [6]. Ces points de fuites nous fournissent des informations essentielles sur l'orientation de

l'observateur dans la scène [5], ainsi que sur la distance focale de la caméra [10]. En combinant cet indice avec le mouvement apparent, on obtient suffisamment d'informations pour pouvoir distinguer les sept mouvements caméra, même lorsqu'ils sont combinés [13].

L'article qui suit présente une nouvelle méthode qui combine le mouvement apparent avec l'information perspective des points de fuites pour la détection du mouvement de la caméra. Il s'agit, à notre connaissance, de la première méthode qui arrive à distinguer tous les mouvements caméra, et à distinguer le zoom du dolly en particulier, ainsi que la combinaison en sens opposé de ces deux mouvements, appelé un «plan trombone».

Cet article fut corédigé par moi-même ainsi que le professeur François Deschênes et ma collègue Wei Pan. En tant que premier auteur, ma contribution à ce travail fut l'essentiel de la recherche sur l'état de l'art, le développement de la méthode, l'exécution des tests de performance et la rédaction de l'article. Le professeur Francois Deschènes, second auteur, a fourni l'idée originale. Il a aidé à la recherche sur l'état de l'art, au développement de la méthode ainsi qu'à la révision de l'article. Wei Pan, troisième auteure, a contribué à la recherche sur l'état de l'art ainsi qu'à l'exécution des tests de performance.

Combining Apparent Motion and Perspective as Visual Cues for Content-based Camera Motion Indexing

Mathieu Marquis-Bolduc[1], François Deschênes[1,2,*], Wei Pan[1]

[1]Département d'informatique
Université de Sherbrooke
2500 boul. Université
Sherbrooke, Qc, Canada, J1K 2R1

[2]Université du Québec en Outaouais
283, boul. Alexandre-Tache
C.P. 1250, succ. Hull
Gatineau, Qc, Canada, J8X 3X7

{Mathieu.Marquis-Bolduc, Francois.Deschenes, Wei.Pan}@usherbrooke.ca

Abstract: We propose a new method for a qualitative estimation of camera motion from a video sequence. The proposed method suggests to use the properties of vanishing point perspective to complete the information obtained from apparent motion, that is to use a cooperative estimation from several visual cues. Focal length and rotational parameters are first retrieved using perspective, then apparent motion is used to retrieve remaining parameters. The proposed method can retrieve all seven camera motions, including combination of motions. Experimental results confirm the usefulness of the additional information gained from perspective.

Keywords: Camera motion parameters estimation, Apparent motion, Optical flow, Vanishing points

* Corresponding Author. Tel.: 1-819-595-3980; fax: 1-819-595-3825

1 Introduction

Content-based indexing and retrieval of video sequences relies on automatic understanding of video content. According to cinematography literature, a scene content is commonly described by three main components [3]. The first one is the subject, which represent what is filmed. It can be anything: a person, an explosion, a building, etc. It is obvious that automatic detection of the subject is difficult. The second one is the distance to the subject. It is divided into a finite number of categories: close shot, full shot, long shot and so forth. It has the advantage of being finite, but the disadvantage of being relative to the subject. Finally, there is the 3D motion of the filming camera, which has a finite number of motion categories. Tracking, booming and dollying respectively describe a translation of the camera (travelling) along its horizontal, vertical and optical (principal) axis. Tilting, panning and rolling describe rotation around the same axis. Together they form the *orientation* information. A change in focal distance is noted by the term zooming. Camera motion conveys a lot of semantic information. Different camera motions on identical scenes can convey a totally different meaning to the audience [3]. For example, a camera panning on a room can convey a sense of *observation*, while a camera travelling through the same room will convey a sense of *exploration*. Moreover, camera motion can be seen as independent of both of the previously mentioned components: a camera rotating on its vertical axis is always called panning, regardless of what is filmed. There are different nomenclatures to describe the various camera motions. Despite this importance, it is not always easy, even to the human eye, to identify the actual 3D motion in any given sequence. It is even more difficult if several motions are combined together, or if the camera intrinsic parameters are changing or unknown.

One of the most intuitive ecue for camera motion recognition is the apparent motion. It corresponds to the projection of relative motion between the camera and the scene onto the image plane. It is also, to our knowledge, the most widely used. Several examples on

how to take advantage of apparent motion have been introduced in the last two decades (see section 2). However, automatic indexing and retrieval should rely on a system that can ideally distinguish between all of the seven basic camera motions in a video sequence, something that, to the authors' knowledge, has not been achieved yet. For example, almost all of the apparent-motion based methods cannot differentiate zoom from dolly, since their apparent motion are almost identical. In addition, most of the time no information is available about the video sequence's camera intrinsic parameters. In order to be suitable to an automatic indexing and retrieval system a camera motion detection scheme should not rely on knowledge of these parameters. Apparent motion by itself does not seems to contain sufficient information to allow full camera motion detection of video sequences with unknown intrinsic parameters. Additional cues or assumptions about the scene are needed to complete the information from apparent motion.

In this paper, we propose a method to detect any single or combined camera motions in a video sequence from both the perspective information, and the apparent motion. The idea is to use perspective information to provide the additional camera and scene information that is needed to recognize all of the camera motions. Our approach is novel in many regards. First, as far as we know, we present the first method that combines the extraction of rotational and focal components from vanishing points and the extraction of translational components from apparent motion of video sequences. Second, the combination of both the apparent motion and the perspective information allows us to identify a greater variety of camera motions. We can discriminate similar motions like pan and track or dolly and zoom, whenever it is possible. Thus, the proposed method improves the accuracy of indexing in an automated content-based indexing and retrieval system. Our approach relies on the common assumption of no object motion. However, experimental results show that in practical situations, this assumption can be broken up to a certain level.

In the next section, we review existing methods to determine camera motion using ap-

parent motion or vanishing points. In section 3, we expose how to combine orientation information from vanishing points with apparent motion information. We also deal with the special case of limited perspective information. In section 4, we present a summary of our camera motion detection algorithm. In section 5, we present a performance evaluation that demonstrates both the usefulness and the efficiency of our method. Finally, we conclude in section 6.

2 Overview of Existing Approaches

Both the apparent motion and the perspective have been extensively used separately as visual cues to perform partial or complete camera motion detection. In this section, we focus on works that are most related to the method we propose, that is, unassisted methods that aim to retrieve camera motion from un-calibrated video sequences. There are two main categories of methods to detect camera motion in a video sequence. The first one includes methods that rely on apparent motion, and more especially on optical flow as an approximation of it. The second one includes methods that rely on perspective information.

2.1 Identifying Camera Motion From Apparent Motion

Apparent motion is the most common cue used to track camera motion information. Assuming that there is no object motion, the apparent motion $(U_{X,Y}, V_{X,Y})$ is given by the following. Note that all symbols in lowercase "x" are in the camera's $\mathbb{R}^3$ coordinate system, with vector (0,0,1) being the camera's optical axis, while those in uppercase "X" are in the image's $\mathbb{R}^2$ coordinate system. All coordinates in the image use the optical center as origin.

$$U_{X,Y} = -\frac{f_x}{z} \cdot t_x + \frac{X}{z} \cdot t_z - \frac{X \cdot Y}{f_x} \cdot r_x - (f_x + \frac{X^2}{f_x}) \cdot r_y + Y \cdot r_z + \frac{\delta f_x \cdot X}{z} + \frac{\delta f_x \cdot X}{f_x} \tag{1}$$

$$V_{X,Y} = -\frac{f_y}{z} \cdot t_y + \frac{Y}{z} \cdot t_z - \frac{X \cdot Y}{f_y} \cdot r_y - (f_y + \frac{Y^2}{f_y}) \cdot r_x + X \cdot r_z + \frac{\delta f_y \cdot Y}{z} + \frac{\delta f_y \cdot Y}{f_y}.$$

where f_x, f_y are the focal distances in x and y pixel units, t_x, t_y, t_z and r_x, r_y, r_z are respectively the translation and rotation of the camera along three orthogonal axes, with z as the camera's optical axis and y axis pointing upward. Finally, z is the scene depth for the image point (X,Y). Note that the image origin is at the optical center. This corresponds to the intersection of the camera optical axis, often termed principal axis, with the image plane. In this equation, the apparent motion caused by camera translation (track (t_x), boom (t_y) and dolly (t_z)) correspond to:

$$U_{X,Y}^{translation} = -\frac{f_x}{z} \cdot t_x + \frac{X}{z} \cdot t_z \tag{2}$$

$$V_{X,Y}^{translation} = -\frac{f_y}{z} \cdot t_y + \frac{Y}{z} \cdot t_z.$$

while the one caused by rotation (pan (r_x), tilt (r_y) and roll (r_z)) is:

$$U_{X,Y}^{rotation} = -\frac{X \cdot Y}{f_x} \cdot r_x - (f_x + \frac{X^2}{f_x}) \cdot r_y + Y \cdot r_z \tag{3}$$

$$V_{X,Y}^{rotation} = -\frac{X \cdot Y}{f_y} \cdot r_y - (f_y + \frac{Y^2}{f_y}) \cdot r_x + X \cdot r_z.$$

Finally, the apparent motion due to zooming (δf) is given by:

$$U_{X,Y}^{zoom} = \frac{\delta f_x \cdot X}{z} + \frac{\delta f_x \cdot X}{f_x} \tag{4}$$

$$V_{X,Y}^{zoom} = \frac{\delta f_y \cdot Y}{z} + \frac{\delta f_y \cdot Y}{f_y}.$$

Equations (1) to (4) are easily obtained by differentiation of the perspective projection equations of a pin-hole camera [32, 22].

Common techniques use equation (1) with optical flow as an estimation of $(U(X,Y), V(X,Y))$ to extract camera motion information [12, 28, 32, 33]. However, certain classes of motion can yield very similar apparent motion. For example, let us compare the apparent motion of tracking and panning.

$$\begin{aligned} U_{X,Y}^{track} &= -\frac{f_x}{Z} \cdot t_x \\ V_{X,Y}^{track} &= \quad 0. \end{aligned} \tag{5}$$

$$\begin{aligned} U_{X,Y}^{pan} &= -(f_x + \frac{X^2}{f_x}) \cdot r_y \\ V_{X,Y}^{pan} &= \quad -\frac{X \cdot Y}{f_y} \cdot r_y. \end{aligned} \tag{6}$$

As can be seen, the apparent motion for tracking is strictly horizontal. In the case of panning, if the focal distance is large(i.e. small field of view) compared to the magnitude of X and Y, then the magnitude of the vertical component of the estimated apparent motion if often negligible compared to the horizontal component. Thus, it can be difficult to distinguish panning from tracking using apparent motion of a scene of unknown shape. An identical conclusion can be drawn for booming and tilting:

$$\begin{aligned} U_{X,Y}^{boom} &= \quad 0 \\ V_{X,Y}^{boom} &= -\frac{f_y}{Z} \cdot t_y, \end{aligned} \tag{7}$$

$$\begin{aligned} U_{X,Y}^{tilt} &= \quad -\frac{X \cdot Y}{f_x} \cdot r_x \\ V_{X,Y}^{tilt} &= (f_y + \frac{Y^2}{f_y}) \cdot r_x. \end{aligned} \tag{8}$$

and for dollying and zooming:

$$U_{X,Y}^{dolly} = \frac{t_z}{Z} \cdot X \tag{9}$$
$$V_{X,Y}^{dolly} = \frac{t_z}{Z} \cdot Y,$$

$$U_{X,Y}^{zoom} = (\frac{\delta f}{Z} + \frac{\delta f_x}{f_x}) \cdot X \tag{10}$$
$$V_{X,Y}^{zoom} = (\frac{\delta f}{Z} + \frac{\delta f_y}{f_y}) \cdot Y.$$

The direction of the apparent motion of dollying and booming is strictly dependant on the relative position of the optical center. Other parameters (depth and focal distance) will only vary the magnitude of the apparent motion vector. This means that it is impossible to tell zoom apart from travelling using only apparent motion, *unless some higher level assumptions are made about the scene depth or shape.* Among all of the seven parameters, only roll can be distinguished from other types of motions without any ambiguities. This means that to recover all of the seven camera motions, we need additional information or constraints to distinguish rotation apart from translation, and dollying apart from zooming.

Most of the methods try to identify camera motions that, combined together, yield an apparent motion field similar to the one computed from the video sequence. As we just demonstrated, the main difficulty of working solely with apparent motion is due to the fact that the projection of different camera motions may result in similar apparent motion. Another difficulty is that the apparent motion equations (see equation (1)) are non-linear when simultaneously taking into account every types of camera motions. To resolve these ambiguities in the case of uncalibrated camera, many of those methods separate motions in groups of visually similar motions [27, 33] or ignore certain motions [12, 28, 32]. These simplifications remove the need for an additional cue, and allow detection of up to six parameters. Others use spatio-temporal information directly [26], that is they find patterns in temporal slices taken from the spatio-temporal volume of the video sequence that correspond to specific camera motions. Motion parallax methods

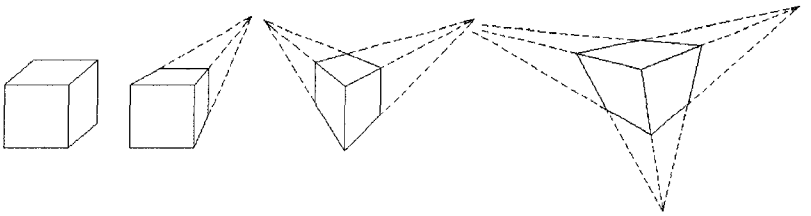


Fig. 1. The box on the left is under orthographic projection. The other boxes are under perspective projection, with one, two and three non-infinite vanishing points.

[16, 20] make assumptions about the presence of certain structures in the scene (e.g planes [16]), that they can register between frames to isolate translation. Methods based on tracking of invariants often assume a calibrated camera (e.g. [29]) to distinguish rotation from translation. Global motion approaches [11, 14, 34] usually aim to separate the apparent motion in 2D classes to predict future apparent motion, rather than identifying the actual 3D motion of the camera. For a more extensive review of apparent motion methods to extract camera motion, please refer to [27]. Based on this review, apparent motion is insufficient to allow unrestricted detection of all of the seven camera motions from a scene filmed with an unknown camera.

2.2 Identifying Camera Motions from Perspective

As previously mentioned, we are interested in using perspective as a visual cue to solve the ambiguities that arise while using apparent motion. When parallel lines in the scene are projected onto the image plane, they may be distorted by perspective and meet at a common image point called vanishing point. This imaginary point is situated at infinity in the 3D scene. If a set of lines is parallel to the image plane, their vanishing point will also be at infinity in the image plane. Assuming that there is a sufficient number of orthogonal and parallel lines in the scene, which is often the case in human made environments, three orthogonal sets of parallel lines can often be retrieved. Depending on camera orientation, such a scene projected using perspective projection will induce 1,

2 or 3 non-infinite vanishing points. These different situations are depicted in figure 1.

Let us first recall what vanishing points are exactly. Given a line following a certain unitary direction (u, v, w) in the scene, passing through a point (x_0, y_0, z_0). Any point on that line can be expressed in homogeneous coordinates as:

$$\begin{pmatrix} x_0 + t \cdot u \\ y_0 + t \cdot v \\ z_0 + t \cdot w \\ 1 \end{pmatrix}. \tag{11}$$

where t is a parametric factor. Using homogeneous coordinate properties we can re-write equation (11) and let t go toward infinity, which gives us:

$$\begin{pmatrix} \frac{x_0}{t} + u \\ \frac{y_0}{t} + v \\ \frac{z_0}{t} + w \\ \frac{1}{t} \end{pmatrix} =_{t \to \infty} \begin{pmatrix} u \\ v \\ w \\ 0 \end{pmatrix}. \tag{12}$$

This is the intersection point, in the scene, of all parallel lines following direction $D = (u, v, w)$, as they tend toward infinity. We can project this point onto the image plane using the 3x4 pin-hole projection matrix $\mathbf{P}$ to find the associated vanishing point $VP_{u,v,w}$

$$VP_{u,v,w} = \mathbf{P} \cdot D = \begin{pmatrix} f_x & 0 & 0 & 0 \\ 0 & f_y & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot (u, v, w, 0)^\top = (u \cdot f_x, v \cdot f_y, w)^\top. \quad (13)$$

The resulting point is the vanishing point associated with the direction (u, v, w), in homogeneous coordinates.

Many methods based on perspective, and more specifically those using a varying number of vanishing points, have been proposed to retrieve shape and/or motion information from static and dynamic scenes. In the single-view case, Gallagher [13] uses the vertical vanishing point, which is usually near to infinity, to correct the effect of camera roll in pictures by aligning the horizon. Cipolla et al. [7] use vanishing points along with user interaction to perform calibration and 3D reconstruction of scenes, by recovering the projection matrices. While the aforementioned are dedicated to single-view images, the same concepts can be ported to video sequences.

For the multiple images case, where the scene is captured from several points of view, most of the work is mainly concentrated around recovering the orientation. In that sense, Martins et al. [23] use a Bayesian probabilistic model to infere camera orientation, using the expected regularity of human environments that provides easily detectable vanishing points. Caprile et Torre [5] propose a method to retrieve camera rotation between two frames. Knowing each frame's orientation by projecting the vanishing points on the unit sphere, the rotation can be retrieved with a matrix inversion. They also propose a method to estimate camera translation if the length of an object in the scene is known, using a triangulation to recover the translation vector. Shigang et al. [31] propose a perspective method to retrieve the orientation of a robot moving on a horizontal plane. Since in their case rotation is around the vertical axis only, they use horizontal lines on the ground

and compute the rotation angle directly. Some methods based on omni-directional images recover the orientation of the cameras using vanishing points, then proceed to extract translation from image correspondences [1, 2]. Those methods are often designed to work best when the camera intrinsic parameters are known and unchanging, and motion parameters very large.

Perspective can also be used to retrieve focal distance information. Daniilidis and Ernst [10] proposed an active method for camera calibration, where the camera is purposefully rotated while looking at a perspective scene to find the camera intrinsic parameters, including the focal distance whose change cause zooming. Several other methods [5, 6, 35] use vanishing points of calibration grids for the same purpose. Kanatani [17] uses two orthogonal vanishing points to retrieve the focal distance in pixel units, achieving partial calibration.

Based on this short review, several methods use either apparent motion or perspective to retrieve camera motion information. It is however clear that they all have limitations that make them inappropriate or difficult to use for automatic indexing of video sequences. Especially, very few, if any, can retrieve all of the 7 basic camera motions. To achieve this goal and work around those limitations, we suggest in this paper to combine apparent motion with perspective information.

3 Identifying Camera Motion From Both Apparent Motion And Vanishing Point Perspective

In what follows, we propose a multi-step approach to estimate all of the seven camera motions. The idea behind our method is to first detect rotation and zoom parameters using perspective information only, and then to infer the remaining parameters from the apparent motion equations. This idea is based on vanishing point properties. In the

proposed method, we relies on the following assumptions:

Single Shot For simplicity, we assume that the sequence is made of a single shot, that is from a single camera, with neither cuts nor transitions. If this is not the case, existing methods to segment camera shots can be used (e.g. [19, 30, 36])

Manhattan World We assume that three vanishing points can be detected, which correspond to three orthogonal directions in the scene.

Optical Flow Computation We assume that optical flow can be computed for a significant part of the frames.

Pixel Ratio We assume that the ratio between pixel's width and height (f_y/f_x) is known or equal 1.

No Object Motion We assume no object motion. If object motion is present, it will act as noise.

Pin Hole Camera We used a pin-hole camera model. We do not take into account the effects of skew, radial distortions, etc.

As can be seen from equation (13), if the direction associated to a given vanishing point is parallel to the image plane i.e. ($w = 0$) in equation (13), the vanishing point is at infinity on the image plane:

$$VP_{u,v,0} = (u \cdot f_x, v \cdot f_y, 0). \tag{14}$$

On the other hand, if this direction corresponds to the optical axis direction (i.e, $u = 0$ and $v = 0$) in equation (13)), then the point is at the optical center $(0, 0)$:

$$VP_{0,0,w} = (0, 0, w) = (0, 0, 1). \tag{15}$$

Another important property that can be observed from these equations is that vanishing point positions on the image plane depend exclusively on both the orientation of the 3D scene lines in the camera's coordinate system, and the camera focal distance. Vanishing

points are thus not influenced by a translation of the camera, since this does not change the orientation of the lines with respect to the camera axis system. This confirms that vanishing points can give us additional information to resolve the ambiguities between equations (5) and (6), between equations (7) and (8), and between equations (9) and (10).

3.1 Retrieving Camera Parameters from Vanishing Points

Lets us take three ($p = 1, 2, 3$) vanishing points in homogeneous coordinates $VP_{p,t} = (X_{p,t}, Y_{p,t}, W_{p,t})$ over two frames t_1 and t_2. From Eq. (13) we know that each vanishing point correspond to a certain scene direction $\overrightarrow{D}_p$. Inversely, using the coordinate system of the first frame as the reference system, we can express $\overrightarrow{D}_p$ in relation to $VP_{p,1}$ as thus:

$$VP_{p,1} = \begin{pmatrix} X_{p,1} \\ Y_{p,1} \\ W_{p,1} \end{pmatrix} = \begin{pmatrix} u_p \cdot f_1 \\ v_p \cdot f_1 \\ w_p \end{pmatrix} \rightarrow \begin{pmatrix} u_p \\ v_p \\ w_p \end{pmatrix} = \begin{pmatrix} \frac{X_{p,1}}{f_1} \\ \frac{Y_{p,1}}{f_1} \\ W_{p,1} \end{pmatrix} = D_p \tag{16}$$

recalling that D_p is a direction and thus has no magnitude per se. For simplification, we assume $f_t = f_x = f_y$. Our experimental results show that this assumption has a negligible impact on parameters estimation. Using the projection matrix of the second frame, $\mathbf{P_2}$, and the rotation matrix between both frames, $\mathbf{R}$, we can express these vanishing points in the coordinate system of the second frame, which gives us three independent constraints:

$$\begin{aligned} VP_{1,2} &= \mathbf{P_2R}(\overrightarrow{D}_1) \\ VP_{2,2} &= \mathbf{P_2R}(\overrightarrow{D}_2) \\ VP_{3,2} &= \mathbf{P_2R}(\overrightarrow{D}_3). \end{aligned} \tag{17}$$

where

$$\mathbf{P}_2 = \begin{pmatrix} f_2 & 0 & 0 & 0 \\ 0 & f_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{18}$$

There are five parameters to retrieve. First, the three rotational parameters from the rotation matrix $\mathbf{R}$. Second, the two focal distances f_1, f_2, which allows us to determine the matrix $\mathbf{P}_2$. As can be seen, the problem is ill-defined since there are 5 unknowns and only 3 constraints. This can be circumvent by assuming that all of the three vanishing points correspond to orthogonal directions. This constraint can be expressed using the dot product of the directions:

$$\overrightarrow{D}_p \bullet \overrightarrow{D}_q = 0, \forall p \neq q. \tag{19}$$

Kanatani [17] first proposed this constraint, sometimes called the "Manhattan World" assumption [9], to estimate the focal distance at each frame. Using the constraint in equation (19) with equation (16), we only have to find the value of f_t that makes vanishing point's directions orthogonal to each other in frame t. As Kanatani shown, solving both equations for two vanishing points i and j yields the following equation:

$$f_t = \sqrt{(-(X_{i,t}X_{j,t} + Y_{i,t}Y_{j,t}))}. \tag{20}$$

Further details on this technique can be found in [17].

Once the focal distance is known for each frame, the rotation parameters can be retrieved from the vanishing points, since we now have four equations (17) and (19), and only three rotational parameters (rotation matrix $\mathbf{R}$) to recover. Since zoom is known and the vanishing points are expressed in the scene 3D coordinate system, as in equation (16), then the rotation parameters can be retrieved using a matrix inversion (Caprile

and Torre [5]):

$$\mathbf{R} = \begin{pmatrix} & & \\ Vp_{1,2} & Vp_{2,2} & Vp_{3,2} \\ & & \end{pmatrix} \begin{pmatrix} & & \\ Vp_{1,1} & Vp_{2,1} & Vp_{3,1} \\ & & \end{pmatrix}^{-1}. \tag{21}$$

Altough other techniques could also be used (cf. section 2), this technique has proven to be simple and efficient by our experiments.

3.2 Retrieving the Translational Parameters from Apparent Motion

Up until now, we retrieved the focal distance at each frame as well as the rotational parameters from the vanishing points. We then approximate the apparent motion using optical flow, and substract the zoom and rotational components (equations (3) and (4)) from equation (1) to obtain a *residual flow*. In other words, we obtain a simplified flow that is only affected by camera translation. Obviously, we rely on the assumption that the depth-related part of the zoom flow in equation (4), is negligible:

$$\frac{\delta f \cdot X}{Z} \simeq 0. \tag{22}$$

We make this assumption since we have *a priori* no knowledge about the scene yet. This introduces a small error in the residual flow. Since Z is generally large compared to f_x, f_y and X, we have experimentally confirmed that this error is negligible and does not influence the ability of the proposed algorithm to distinguish all of the seven camera motions. The residual flow is then:

$$\begin{aligned} U_{X,Y}^{residual} &= -\frac{f_x}{Z} \cdot t_x + \frac{X}{Z} \cdot t_z \\ V_{X,Y}^{residual} &= -\frac{f_y}{Z} \cdot t_y + \frac{Y}{Z} \cdot t_z. \end{aligned} \tag{23}$$

This residual flow contains none of the ambiguities of the original flow: All of the three remaining components have very different apparent motion, and we can be retrieved from optical flow using a minimization scheme.

Notice that dollying may however locally look like a mix of tracking and booming. In order to circumvent this problem and improve results, we use a technique similar to the one proposed by Srinivasan et al. [32]. They argue that if all flow is removed except the part caused by tracking and booming, the residual flow should be parallel in the image plane, and they attempt to find the camera rotation and zoom parameters that will make the residual flow parallel. Since we already removed the flow caused by rotation and zoom, we propose to find the translation parameters that will make the residual flow at each point go directly toward or away from the optical center, like the flow of dolly, or that will simply minimize the residual flow. This approach is especially convenient in our particular case since there is no need to estimate the flow alignment at each step of the minimization process (see [32]). The parameters to be estimated are the global translation parameters t_x, t_y and the scene depth z at all points that are considered. Given the following definitions:

$$\begin{aligned} U_{X,Y}^{min} &= U_{X,Y}^{residual} + \frac{f_x}{z} \cdot t_x = U_{X,Y}^{dolly} \\ V_{X,Y}^{min} &= V_{X,Y}^{residual} + \frac{f_y}{z} \cdot t_y = V_{X,Y}^{dolly}. \end{aligned} \tag{24}$$

and

$$\theta_{X,Y}^{1} = \arccos \frac{X}{\sqrt{X^2 + Y^2}} \tag{25}$$

$$\theta_{X,Y}^{2} = \arccos \frac{U_{X,Y}^{min}}{\sqrt{(V_{X,Y}^{min})^2 + (U_{X,Y}^{min})^2}}. \tag{26}$$

we attempt to minimize the total angular error. $\theta^1_{X,Y}$ is the orientation of the (X,Y) image point, relative to the optical center. $\theta^2_{X,Y}$ is the orientation of the minimized optical flow $(U^{min}_{X,Y}, V^{min}_{X,Y})$ relative to this point. It can be seen that if the minimized flow contains only dollying (c.f. equation (9)), then both orientations d be identical. Our cost function for the minimization is thus:

$$Cost = \sum_{X,Y} (\theta^2_{X,Y} - \theta^1_{X,Y})^2. \tag{27}$$

In some cases, an infinity of combination of t_x and t_y could minimize the angular error. This is due to the fact that we only considered the orientations of the vectors and not their magnitudes. To avoid this ambiguity, we include the X and Y translation magnitude in the minimization. In the case of an infinity of equivalent solutions having different norm, the minimization will thus settle for the one with the lowest magnitude. In most cases including dolly, experiments confirm that this additional part to the cost function do not influence the results. α and β are used to weight both parts of the final constraint equation:

$$Cost = \sum_{X,Y} (\alpha(\theta^2_{X,Y} - \theta^1_{X,Y})^2 + \beta(t_x^2 + t_y^2)). \tag{28}$$

We use two weighting constants instead of one to help avoid potention numerical problems, however, a single one could be used. Notice that if there is significant tracking or booming, we also retrieve the scene depth z up to a scale factor. One parameter still need to be evaluated: dollying (t_z). It can be computed directly using a least-square minimization of equation (9). This also leads to a simultaneous estimation of both t_z and z. Notice that if the scene depth (Z) was recovered at the previous step, it can be used to improve the estimation of dollying. In other words, the only unknown in that case is t_z.

We have shown how to retrieve all of the seven motion parameters, and also the scene depth when at least one of the translation parameters is significant. Recall that it was

based on the assumption that two or three of the orthogonal vanishing points are not located at infinity in the image plane. If this is not the case, then it might also be possible to estimate all of the camera motion parameters, as will be shown in the next section.

3.3 Particular Case: Single Vanishing Point

As can be seen from equation (13), if the direction associated to a given vanishing point is parallel to the image plane ($w = 0$), then the vanishing point is located at infinity in the image. If all of the frames contain only one vanishing point that is not at infinity, then it can be assumed that there is neither camera panning nor tilting, since any presence of pan or tilt would quickly switch the scene to the two or three-point vanishing point cases. In this case the remaining vanishing point will be located at the optical center $(0, 0)$, according to the vanishing point orthogonality assumption from Eq. (19). Notice that this situation is common in practice. For example, most of existing movies and numerous T.V. programs contain at least one shot of this type. We must also mention that when two vanishing points are at infinity and the remaining one is at the image plane origin, they do not convey any information about focal distance. Concerning roll, it can be easily estimated using the orientation of the two vanishing points at infinity, since a roll of r_z degrees will affect the position of those vanishing points around the optical center by exactly the same amount (i.e. r_z degrees). Note that unlike pan and tilt, estimating roll in this way is regardless of the focal distance. Given the position $X_{p,t}, Y_{p,t}$ in the image of vanishing points ($p = 1, 2$) located at infinity at frame t, we can express the orientation of those vanishing points using:

$$\theta_{p,t} = arccos(\frac{X_{p,t}}{\sqrt{X_{p,t}^2 + Y_{p,t}^2}}). \tag{29}$$

The camera roll can then be estimated using the differences between $\theta_{1,1}$ and $\theta_{1,2}$ and/or $\theta_{2,1}$ and $\theta_{2,2}$. For example, one can use the mean of those differences:

$$r_z = \frac{(\theta_{1,1} - \theta_{1,2}) + (\theta_{2,1} - \theta_{2,2})}{2}. \tag{30}$$

As for the tracking and booming parameters, since zooming and dollying have the same orientation, they can be estimated using the minimization scheme proposed in the previous section. In this case, the residual flow is given by:

$$\begin{aligned} U_{X,Y}^{residual} &= -\frac{f_x}{z}t_x + \frac{X}{Z}t_z + \frac{\delta f_x X}{z} + \frac{\delta f_x \cdot X}{f_x} \\ V_{X,Y}^{residual} &= -\frac{f_y}{z}t_y + \frac{Y}{Z}t_z + \frac{\delta f_y Y}{z} + \frac{\delta f_y \cdot Y}{f_y}. \end{aligned} \tag{31}$$

We can obtain the residual flow by substraction Eq. (3) from the optical flow in input. Unfortunately, the residual optical flow equations do not yield linear constraints. We then suggest to use a non-linear minimization scheme using the apparent motion related to both zoom and dolly :

$$\begin{aligned} U_{X,Y}^{min2} &= U_{X,Y}^{residual} + \frac{f_x}{z}t_x \\ V_{X,Y}^{min2} &= V_{X,Y}^{residual} + \frac{f_y}{z}t_y, \end{aligned} \tag{32}$$

$$\begin{aligned} A_{X,Y} &= U_{X,Y}^{min2} - \frac{X}{z}t_z - \frac{\delta f_x X}{z} - \frac{\delta f_x \cdot X}{f_x} \\ B_{X,Y} &= V_{X,Y}^{min2} - \frac{Y}{z}t_z - \frac{\delta f_y Y}{z} - \frac{\delta f_y \cdot Y}{f_y}. \end{aligned} \tag{33}$$

We attempt to minimize the sum of square differences between the flow produced by the estimated parameters and the actual residual flow:

$$E_2 = \sum_{X,Y}(A_{X,Y}^2 + B_{X,Y}^2). \tag{34}$$

Under our previous assumption that $f_x = f_y$, minimizing E_2 lead to a simultaneous estimation of zoom (δf), dolly (t_z). Scene depth (z) is also estimated if, and only if, dolly is present. Notice that the two U, V components for zoom and dolly are not independent. For a fixed point X, Y, the horizontal and vertical optical flow of these motions are dependent. Thus, for each point, we only have one equation, i.e. Eq. (34). If the depth is unknown (i.e. whenever the scene contains no dolly), then for N points we have $N + 2$ unknowns and only N equations, which makes the system underconstrained. Fortunately, we can use lines converging toward the existing vanishing point (vanishing lines) to circumvent this problem. From equation (11), we can assume that the depth of points on lines that converge toward the optical center (the only finite vanishing point in that case) will increase as the point is closer to the optical center. It is thus possible to build an approximate depth map, up to a scale factor, for the parts of the image along those lines. Our system is now solvable, and we can retrieve zoom, dolly and the focal distance, up to a scale factor.

4 Summary of the Algorithm

Let us now present the resulting algorithm based on previous developments. We will then present some details in order to facilitate the implementation.

4.1 Overview of the Algorithm

4.1.1 Underlying assumptions

This work relies on common assumptions:

Single Shot For simplicity, we assume that the sequence is made of a single shot, that is from a single camera, with neither cuts nor transitions. If this is not the case, existing

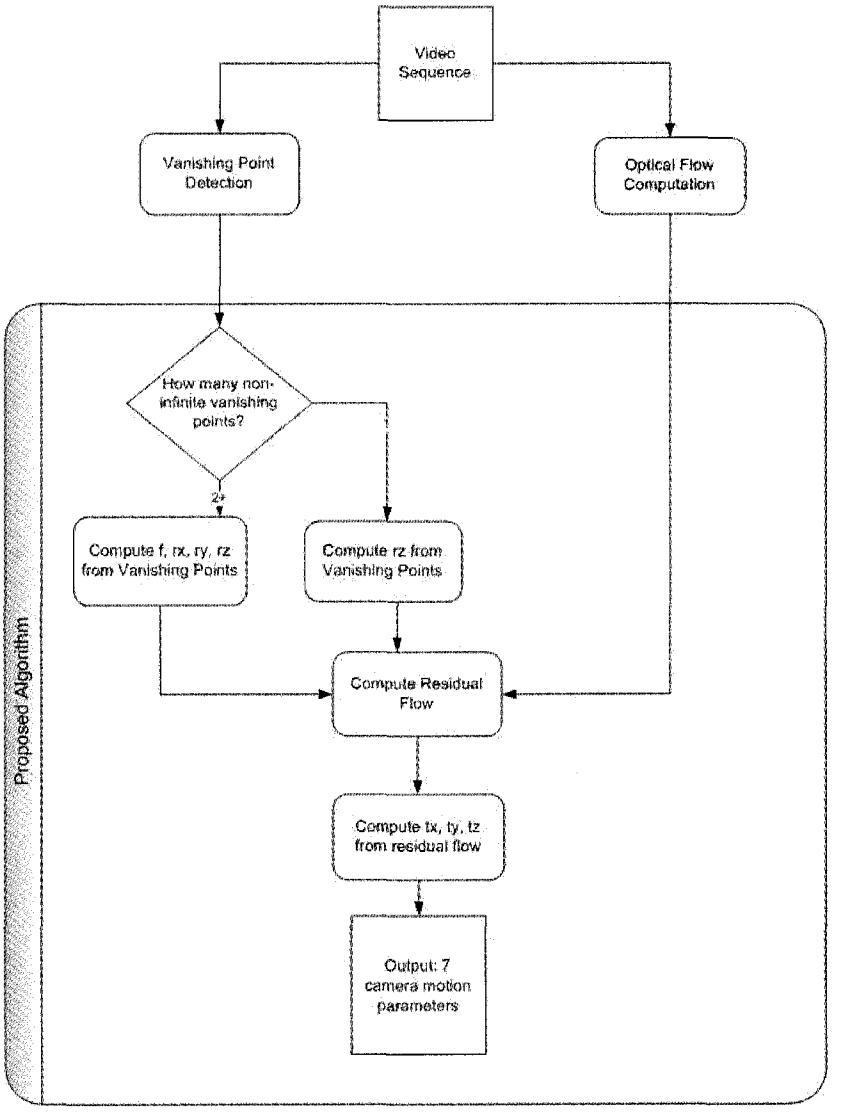


Fig. 2. Overview of the algorithm for camera motion extraction based on apparent motion and vanishing points.

methods to segment camera shots can be used (e.g. [19, 30, 36])

Manhattan World We assume that three vanishing points can be detected, which correspond to three orthogonal directions in the scene.

Optical Flow Computation We assume that optical flow can be computed for a significant part of the frames.

Pixel Ratio We assume that the ratio between pixel's width and height (f_y/f_x) is

known or equal 1.

No Object Motion We assume no object motion. If object motion is present, it will act as noise.

Pin Hole Camera We used a pin-hole camera model. We do not take into account the effects of skew, radial distortions, etc.

Notice that, as will be shown in Section 5, violation of the last three assumptions do not degenerate the results

4.1.2 Input

Our method takes as input the optical flow field of a video sequence as an approximation of the apparent motion. To this end, we can use existing methods, either completely or partially dense, such as [15, 21]. We also take as input three orthogonal vanishing points for each frame. To this end, an existing technique such as [4, 8, 18, 24] can be used. The only parameters of our method are the weights of the minimization scheme (Eq.(28)).

4.1.3 Outline of the Algorithm

As shown in figure 2, the main steps of the algorithm are:

(1) Estimate f for each frame using equation (20).

(2) Estimate r_x, r_y and r_z using equations (21) or (30), depending on the number of vanishing points that are not located at infinity.

(3) Compute the residual flow by subtracting the rotation and zoom flow using equations (3) and (4) from the estimated optical flow.

(4) Estimate t_x, t_y and Z using equation (28).

(5) Compute the second residual flow by subtracting the tracking and booming flow

using equations (5) and (7).

(6) Compute t_z using equation (9) or t_z and δf using equation (34).

4.1.4 Output

The output of the algorithm are the zooming parameter, as a change of the focal distance in pixel units (f_x, f_y), the three rotational parameters of the camera rotation matrix $\mathbf{R}$, and the three translational components. Recall that the value of the translational parameters is not absolute, but relative to a scale factor with scene depth (see Eq. (2)). The particular case of two vanishing points at infinity, presented in section 3.3, is also taken into account. In this case, it is assumed that there is neither pan nor tilt. Zoom is retrieved but up to a scale factor with the translational parameters.

4.2 Implementation Details

4.2.1 One vs Several Vanishing Points Perspective

To evaluate if a vanishing point can be considered at infinity, we use two maximum thresholds (e_x, e_y) related to the image width and height respectively. The closer a vanishing point is to infinity, the less reliable its position are. The farther from the optical center the vanishing point is, the more parallel the lines converging to this point become, and the computation of their intersection is less reliable. Thus, every vanishing point (x, y) for which $x > e_x \cdot I_{width}$ or $y > e_y \cdot I_{height}$ is considered at infinity. These thresholds are applied in image space, and are thus independent of the scene's depth.

4.2.2 Jittering and Vanishing Point Filtering

A common problem occurring in vanishing point detection over a video sequence is jittering. In other words, the estimated vanishing point positions in time are distributed

all around the ground truth. This phenomenon can be caused by an unstable camera or by imprecision in the vanishing point detection. To overcome this problem, we suggest to optionally include a smoothing curve fitting, such as fitting the points on a bezier curve, to align the vanishing point positions before proceeding. This will ensure more consistent results when using noisy vanishing points.

4.2.3 Optical Flow Computation and Temporal Sampling

The optical flow can be computed using any technique the reader considers reliable. However notice that parts of the optical flow where the magnitude is close to zero should be ignored. Either the flow could not be resolved at those points, or the scene depth is too great (possibly at infinity) and this will cause problems for the estimation of both the translation and the depth.

4.2.4 Minimization

To perform the actual minimization of Eq (28) and (34), several techniques could be used. Since our goal was not to evaluate the pros and cons of those techniques we use an actual implementation of the Nelder-Mead's [25] method for experimental purpose. This method has proven to be efficient in prior work (e.g. Srinivasan et al [32]) and does not require the computation of the derivatives of the cost function. Notice that care should be taken to have a good initialization for the Nelder-Mead algorithm using a set of sufficiently large and diverse initial values. The initial values for depth can be set arbitrary without loss of generality, as well as the value of f for the case of a single vanishing point not at infinity, since we will obtain a result up to a scale factor.

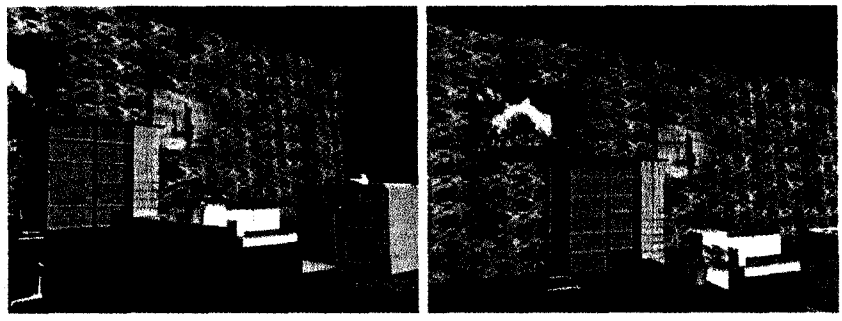


Fig. 3. Frames from the three-motion synthetic sequence at time $t = 0$ and t = 14.

5 Performance Evaluation

In this section we present experimental results of camera motion estimation for a variety of real and synthetic video sequences of architectural scenes. For all of those results we used $\alpha = 1$ and $\beta = 0.01$ as weight parameters (equation (28)), as well as thresholds $e_x = e_y = 25$ in order to determine if a vanishing point is to be considered at infinity. Recall that we use the estimated motion parameters to label frames of the video sequences in order to facilitate indexing and retrieval.

5.1 Evaluation of the Usefulness of the Perspective Cue

In order to evaluate how important is the perspective cue on the estimation of the motion parameters, we compare the results obtained using the proposed method to results obtained from running a similar method based only on apparent motion [32].

5.1.1 Results from Synthetic Sequences

In this section, we present results obtained for a sample of 12 sequences with known ground truth. Each frame is made of 320x200 pixels. A subset of frames is presented in figure 3. This particular sequence is a combination of simultaneous panning, booming and zooming. Sequences contain 60 frames each.

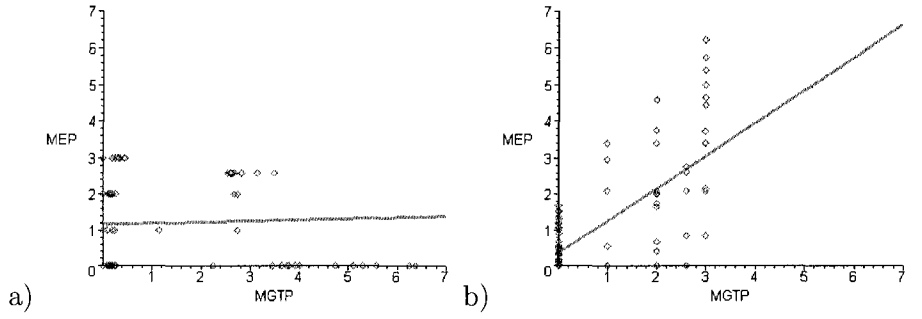


Fig. 4. Correlation between the set of ground truth parameters (MGTP) and the set of estimated parameters (MEP) for all of the seven motions. a) Results from apparent motion only. b) Results from both apparent motion and perspective. Note that the darker dots indicate that several data points are situated at almost the same location on the graph.

Figure 4 shows correlation graphs between absolute value of the estimated parameters (MEP) and the absolute value of ground truth parameters (MGTP) obtained both without and with perspective cue. The plot shows the absolute value of every individual ground truth parameters versus the absolute value of the corresponding individual estimated parameters. Because each frame of each video sequence potentially contains 8 parameters, each frame tested correspond to 8 points in the graph. A line has been fit on data using a least-square regression. Recall that for such graphs, a line with a slope close to 1.0 indicates a good correlation. This means that the estimated parameters are, on average, directly proportional to the ground truth. A very high or very low positive slope indicate a poor correlation, which means that the magnitude of the estimated parameters is unrelated to the magnitude of the ground truth. Finally, a correlation with a negative slope indicates that the estimated parameters magnitude is inversely proportional to the ground truth. Thus as shown in figure 4-a, the magnitude of the motion parameters evaluated by the method based on motion only is unrelated to the magnitude of the ground truth parameters since fitted line has a slope close to zero. In other words, that method fails to accurately identify motion parameters, most of the estimated values being close to zero. This is not surprising as it is difficult, even to the human eye, to identify such a combination of motions, especially when all seven camera motions are present.

In particular, camera panning and camera tracking are indistinguishable. However, as shown in figure 4-b, the use of the vanishing points drastically improves the results. The estimated parameters are more accurate since they correlate with a slope of 0.90. This indicates that with the proposed method, the magnitude of the estimated parameters is, on average, proportional to the magnitude of the ground truth parameters, even when several types of motions occur at the same time. Similar results were obtained for all of the experiments made with sequences involving at least camera rotation or dolly. Notice that, in the results that are presented in Fig. 4, both methods performed well on the subset of sequences involving only camera track, boom or zoom. The computational time needed to extract motion parameters is approximately 1 second for each pair of frame on a 2.2 Ghz computer and depends mainly on the performance of the minimization scheme used.

5.1.2 Results on Real Video Sequences

We now present results of camera motion estimation for 19 real video sequences extracted from movies. One of the movies is Alfred Hitchcock's 1958 *Vertigo*, which was the first to feature a "trombone shot" camera motion, a combination of zoom and dolly. In addition, notice that the camera motion of shots from this movie are usually rotation rather than translation. Another well-known movie we used is Stanley Kubrick's 1980 *The Shining*. This movie was almost entirely shot using a stabilized shoulder-mounted travelling camera. Most of the shots include object motion. Test sequences contain 50 frames in average.

For all of those video sequences, the qualitative accuracy of the results was evaluated using two metrics : sensitivity and specificity. The sensitivity is defined as the average, for each pair of frames, of correctly identified camera motions divided by the number of motions present in the current pair of frames. The specificity is defined as the average, for each pair of frames, of correctly undetected motions divided by the total number of

motion types (7) minus the number of motions that are present in the current pair of frames. To determine if a motion is present, we compare the magnitude of the apparent motion it caused to the magnitude of the optical flow of the frame. Motions with a flow magnitude of at least 10% of the whole are considered present. Tables 1 and 2 summarize the results for all of the real sequences, sorted by the motion(s) present in the sequences. If a sequence contains several motions, it will be accounted in all appropriate categories.

Detected Camera Motion	Sensitivity using our method	Specificity using our method	Sensitivity using other method	Specificity using other method
Zoom	100%	48.73%	100%	96.23%
Tracking	85.63%	24.26%	96.64%	91.5%
Booming	94.25%	31.24%	97.74%	90.84%
Dollying	0%	100%	92.84%	93.85%
Panning	2.33%	80.82%	92.57%	76.76%
Tilting	7.23%	70.23%	90.37%	83.11%
Roll	3.84%	83.72%	94.82%	70.71%

Table 1

Individual camera motion detection results for real video sequences

In section 2.1, we showed the ambiguities that arise when using only the apparent motion to determine camera motion. These ambiguities are well reflected in our experimental results. There is no doubt that the results using both perspective and apparent motion are globally better. Given the unknown aperture angle, the method based only on apparent motion struggles with sequences involving camera rotation. Also, its inability to

Detected Camera Motion	Sensitivity using our method	Specificity using our method	Sensitivity using other method	Specificity using other method
Translation only	100%	15.62%	98.31%	99.15%
Rotation only	4.55%	83.72%	91.48%	72.84%
Translation and Rotation	4.55%	27.48%	93.75%	84.32%
Translation and Zoom	50%	50%	100%	100%

Table 2

Combined camera motion detection results for real video sequences

distinguish dollying from zooming affects the results in most sequences involving dolly. This flaw is responsible for its low score in the "Translation Combined with Rotation" category in table 2. This seems to be related to the fact that dollying is frequently combined with panning in movies [3]. By opposition, we see that the sensitivity rate of the method based on both perspective and apparent motion is very satisfying: the main motions of every sequences were easily identified by our method, even when several type of motions are combined together. The specificity rate decreases in sequences with strong camera rotation, but results are still more satisfying than those obtained from apparent motion only. Please notice that the estimation of rotational parameters is sensitive to small numerical errors and noise that may affect the input. Experiments show that such errors can yield a significant error level in the estimation of rotation parameters. Moreover, error in estimating the rotation between two frames will most likely results in a false translation detection, as the algorithm compensate for the error in the following steps. Thus, accurate vanishing point detection and camera rotation estimation will most likely guarantee a lower false positive rate.

Overall, the experiments confirm that the additional information provided by perspective allows our algorithm to accurately detect all of the seven camera motions. It thus provides an efficient way for labelling video frames in the context of content-based video retrieval.

5.2 Particular Results from a Trombone Shot

Let us now emphasize on a specific example which illustrates the power of our algorithm. The example comes from the "trombone shot" of the stairway scene from the movie *Vertigo*. Figure 5 presents three frames of this shot (in which there is a combination of zoom and dolly), and Figure 6 presents the corresponding computed optical flow. This is a typical single vanishing point perspective case, with two vanishing points at infinity and the last one at the optical center. Note that there is no object motion. The algorithm does not detect any panning, tilting, rolling, tracking or booming, which is correct. A combination of zooming and dollying was correctly identified at each frame. We also note that, as expected, the zooming and dollying parameters signs are always the same and always opposed, e.g., dolly in and zoom out. Table 3 summarizes the results for this sequence. To the authors' knowledge, the proposed method is the first to successfully recognize trombone shots.

6 Conclusion

We presented an algorithm which extract all of the seven camera motions from a video sequence. Our approach successfully combines information from apparent motion with information from vanishing points. First, we use perspective to extract focal length and rotational parameters. These parameters are then used to facilitate the extraction of translational parameters from apparent motion. The performance of our method was evaluated on both synthetic and real sequences. The results are quite satisfying. They

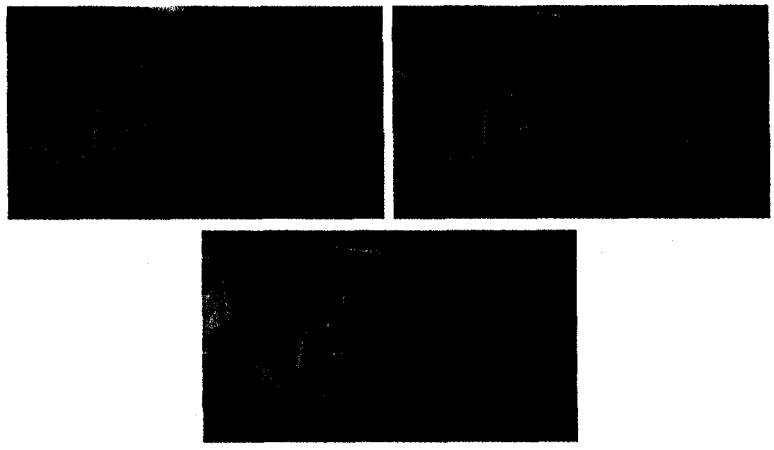


Fig. 5. Frames from the stairway sequence at time t = 0, 10 and 20.

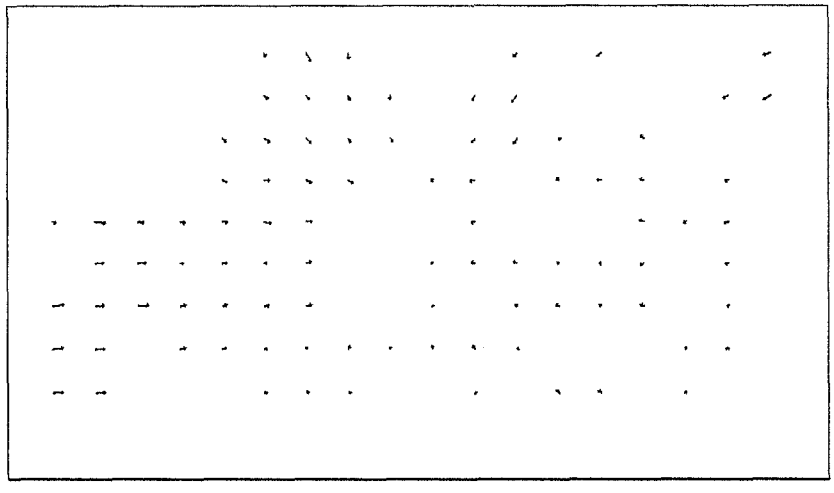


Fig. 6. Example of optical flow obtained at t=10 from the stairway sequence.

confirm that the estimation of the motion parameters is good enough to allow a robust qualitative detection of all of the camera motions. Moreover, the usefulness of perspective information for camera motion detection has been demonstrated by comparing results obtained with our approach to results obtained from a similar approach based on apparent motion only.

Different motions and combinations of motions have different semantic meanings. In that regard, the detection of all of the seven different motions, including combinations

Motion	Sensitivity	Specificity
Zooming	**100%**	**100%**
Panning	N/A	100%
Tilting	N/A	100%
Rolling	N/A	100%
Tracking	N/A	100%
Booming	N/A	100%
Dollying	**100%**	**100%**

Table 3

Camera motion detection result for a trombone shot from the movie *Vertigo*.

of motions, is very important to content-based indexing and retrieval. In that sense, we believe the proposed method has proven to be efficient for identifying most combinations of motions used in cinematography, including the trombone shot.

7 Acknowledgements

This work is partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) research funds and the Fond Québécois de la Recherche sur la Nature et les Technologies (FQRNT).

References

[1] M.E. Antone and S. Teller. Automatic recovery of relative camera rotations for urban scenes. In *Proceedings of the Conference on Computer Vision and Pattern*

Recognition (CVPR '00), volume 2, pages 282–289, Hilton Head Island, United States of America, June 13-15th 2000. IEEE Computer Society.

[2] M.E. Antone and S. Teller. Scalable extrinsic calibration of omni-directional image networks. *International Journal of Computer Vision*, 49(2-3):143–174, 2002.

[3] D. Arijon. *Grammar of the Film Language*. Silman-James Press, 1976.

[4] V. Cantoni, L. Lombardi, M. Porta, and N. Sicard. Vanishing point detection: Representation analysis and new approaches. In *11th International Conference on Image Analysis and Processing*, pages 90 – 94, Palermo, Italy, September 26-28 2001.

[5] B. Caprile and V. Torre. Using vanishing points for camera calibration. *International Journal of Computer Vision*, 4(2):127–140, 1990.

[6] W. Chen and B.C. Jiang. 3-d camera calibration using vanishing point concept. *Pattern Recognition*, 24(1):57–67, 1991.

[7] R. Cipolla, T. Drummond, and D. Robertson. Camera calibration from vanishing points in images of architectural scenes. In T. Pridmore and D. Elliman, editors, *Electronic Proceedings of the Tenth British Machine Vision Conference (BMVC 99)*, volume 2, pages 382–391, Nottingham, United Kingdom, 1999.

[8] R. Collins and R. Weiss. Vanishing point calculation as a statistical inference on the unit sphere. In *International Conference on Computer Vision*, pages 400–403, December 1990.

[9] J. Coughlan and A. Yuille. The manhattan world assumption: Regularities in scene statistics which enable bayesian inference. In *Neural Information Processing System (NIPS)*, pages 845–851, 2000.

[10] K. Daniilidis and J. Ernst. Active intrinsic calibration using vanishing points. *Pattern Recognition Letters*, 17(11):1179–1189, 1996.

[11] F. Dufaux and J. Konrad. A robust, efficient, and fast global motion estimation method from mpeg compressed video. In *Proceedings of the Third IEEE Pacific Rim Conference on Multimedia*, pages 151–158, Hsinchu, Taiwan, December 16-18 2002.

[12] R. Ewerth, M. Schwalb, P. Tessmann, and B. Freisleben. Estimation of arbitrary camera motion in MPEG videos. In *Proceedings of the 17th International Conference on Pattern Recognition (ICPR)*, volume 1, pages 512 – 515, August 23-26 2004.

[13] A.C. Gallagher. Using vanishing points to correct camera rotation in images. In *Proceedings. the 2nd Canadian Conference on Computer and Robot Vision (CRV'05)*, pages 460 – 467, Victoria, Canada, May 9-11 2005.

[14] G. Giunta and U. Mascia. Estimation of global motion parameters by complex linear regression. *IEEE Transactions on Image Processing*, 8(11):1652–1657, 1999.

[15] B.K.P. Horn and B.G. Schunck. Determining optical flow. *AI Memo*, (572), 1980.

[16] M. Irani, B. Rousso, and S. Peleg. Recovery of ego-motion using region alignement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3):268–272, 1997.

[17] K. Kanatani. *Geometric computation for machine vision.* Oxford University Press, Inc., New York, NY, USA, 1993.

[18] J. Kosecka and W. Zhang. Efficient computation of vanishing points. In *IEEE International Conference on Robotics and Automation (ICRA 2002)*, pages 223–228, 2002.

[19] S. Lawrence, D. Ziou, M.F. Auclair-Fortier, and S. Wang. Motion insensitive detection of cuts and gradual transisitions in digital videos. *Pattern Recognition and Image Analysis*, 14(1):109–119, 2004.

[20] M.I.A. Lourakis, A.A. Argyros, and S.C Orphanoudakis. Independant 3d motion detection using residual parallax normal flowfields. In *Proceedings of the Sixth International Conference on Computer Vision*, pages 1012–1017, Bombay, India, January 4-7 1998.

[21] B.D Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision (ijcai). In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, pages 674–679, Vancouver, British Columbia, August 24-28 1981.

[22] J. Ma and S. I. Olsen. Depth from zooming. *Journal of the Optical Society of America A*, 7:1883–1890, 1990.

[23] A. Martins, P. Aguiar, and M. Figueiredo. Navigating in manhattan: 3d orientation from video without correspondences. In *Proceedings IEEE International Conference Image Processing*, volume 1, pages 285–288, Barcelona, Spain, September 14-17th 2003.

[24] G.F. McLean and D. Kotturi. Vanishing point detection by line clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(11):1090 – 1095, November 1995.

[25] J.A Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7:308–313, 1965.

[26] C.-W. Ngo, T.-C. Pong, H.-J. Zhangz, and R. T. Chin. Motion characterization by temporal slices analysis. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR '00)*, pages 768–773, Hilton Head Island, United States of America, June 13-15th 2000. IEEE Computer Society.

[27] W. Pan. *Interpreting Camera Operations in the Context of Content-Based Video Indexing and Retrieval, M.Sc. Thesis.* Université de Sherbrooke, 2006.

[28] S.C. Park, H.S. Lee, and S.W. Lee. Qualitative estimation of camera motion parameters from the linear composition of optical flow. *Pattern Recognition*, 37(4):767–779, April 2004.

[29] B. Rousso, S. Avidan, A. Shashua, and S. Peleg. Robust recovery of camera rotation from three frames. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 796–802, Washington, DC, USA, 1996.

[30] B. Shahraray. Scene change detection and content-based sampling of video sequence. *Digital Video Compression: Algorithms and Technologies*, pages 2–13, february 1993.

[31] L. Shigang, S. Tsuji, and M.S. Imai. Determining camera rotation from vanishing points of lines on horizontal planes. In *Third International Conference on Computer Vision (ICCV90)*, pages 499–502, Osaka, Japan, December 4-6th 1990.

[32] M.V. Srinivasan, S. Venkatesh, and R. Hosie. Qualitative estimation of camera motion parameters from video sequences. *Pattern Recognition*, 30(4):593–606, April 1997.

[33] G. Sudhir and J.C.M. Lee. Video annotation by motion interpretation using optical flow streams. *Visual Communication and Image Representation*, 7(4):354–368, December 1996.

[34] T. Vlachos. Simple method for estimation of global motion parameters using sparse translational motion vector fields. *Electronics Letters*, 34(1):60–62, 1998.

[35] L.L. Wang and W.H. Tsai. Camera calibration by vanishing lines for 3-d computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):370–376, 1991.

[36] H.J. Zhang, A. Kankanhalli, and S.W. Smoliar. Automatic partitioning of full-motion video. *Multimedia Systems*, 1(1):10–28, 1993.

Vitae

M. Marquis Bolduc, B.Sc, is currently a M.Sc. student in Computer science at the Université de Sherbrooke (Canada). His research interests concerns video processing and more especially motion interpretation and 3D scene reconstruction.

F. Deschênes received a Ph.D. degree in Computer science (2002) from both the Ecole Nationale Supérieure des Mines de Paris (France) and the Université de Sherbrooke (Canada). He is currently the Dean of research at the Université du Québec en Outaouais (Canada). He is also Professor at Departement of Computer science at the Université de Sherbrooke since 2002. His research interests mainly concern computer vision and more specifically 3D scene understanding, depth cue extraction and video processing.

W. Pan received a M.Sc. degree in Computer science from Université de Sherbrooke (Canada) in 2006. Her research interests include content-based indexing and retrieval, video processing and image understanding.

Chapitre 2

Occultations dans le volume spatio-temporel

2.1 Avant-Propos

Dans une séquence vidéo, lorsque le mouvement de la caméra et/ou le mouvement des objets fait en sorte qu'un objet dans la scène en occulte un autre, on dit qu'il y a occultation. De même, ces mouvements peuvent occasionner l'apparition graduelle d'un objet qui était occulté. Le type d'événements (occultation et «désoccultation») implique la présence de contours d'occultation, qui sont la projection dans l'image des limites physiques d'un objet dans la scène [9].

La détection des occultations, qu'il s'agisse des événements eux-mêmes ou des contours particuliers qui leur sont associés, est d'un grand intérêt en vision artificielle. D'une part, les occultations présentent un obstacle à certaines méthodes de vision artificielle, tel le calcul du flot optique ou la reconstruction de scène basée sur plusieurs images [9]. D'autre part, elles fournissent aussi des indices importants sur la structure d'une scène

et le comportement des objets contenus dans la scène. Par exemple, les occultations permettent d'ordonner des surfaces ou des objets selon leur distance relative à la caméra [22], aidant ainsi à la reconstruction de la scène [9], ou à la description sémantique de la scène (l'homme passe derrière le mur), ce qui peut se révéler utile dans le domaine de l'indexation automatique par le contenu.

La méthode proposée pour répondre à ce problème est basée sur l'analyse du volume spatio-temporel. Aussi appelé videocube [18], le volume spatio-temporel est composé de l'agrégation d'une séquence d'images continues en un volume compact 3D (x, y et t ; temps). Plusieurs méthodes utilisent le volume spatio-temporel dans d'autres contextes [8, 12, 15, 17, 19] . Dans un tel volume, les contours 2D présents dans les images forment des surfaces, appelées surfaces spatio-temporelles. Ces surfaces peuvent être localisées dans le volume, à l'aide d'un détecteur de contours basé sur le gradient 3D par exemple [4]. L'orientation de ces surfaces dépend à la fois de l'orientation du contour dans l'image ainsi que de la vélocité de son déplacement, s'il y a lieu. Lorsque deux contours se rencontrent en un point donné, les surfaces spatio-temporelles correspondant à ces contours se rencontrent aussi, formant une arête dans le volume spatio-temporel, ce qui correspond à un phénomène d'occultations. Il est possible de détecter les arêtes dans le volume spatio-temporel en utilisant une analyse géométrique, en extrayant par exemple les maximums locaux de la courbure 3D [14]. Dans cette optique, la méthode détaillée dans l'article qui suit propose d'identifier les occultations par la détection et le filtrage des arêtes dans le volume spatio-temporel correspondant à l'intersection de surfaces spatio-temporelles. La méthode décrite permet également d'identifier le contour passant devant l'autre. Ceci fournit une information sémantique additionnelle, discernant l'objet qui est le plus proche de la caméra de celui qui est plus loin.

L'avantage d'une telle méthode est qu'il s'agit d'un procédé uniquement géométrique. La méthode proposée ne nécessite donc au préalable aucune des informations sur la scène ou

la caméra, et ne requiert pas le calcul du flot optique ni celui de la profondeur de scène, qui peuvent se révéler problématiques dans les zones de l'image situées près des contours d'occultation.

L'article qui suit, intitulé «Occlusion Event Detection using Geometric Features in Spatio-temporal Volumes», décrit cette méthode. Cet article a été soumis à la revue *Machine Vision and Applications* en août 2007. Une version courte a été présentée à la conférence *Canadian Conference on Computer and Robot Vision* en 2005 [4]. Il a été corédigé par moi-même ainsi que le professeur François Deschênes. Le professeur Deschênes a fourni l'idée originale et a contribué au développement de la méthode ainsi qu'à la révision de l'article. Ma contribution en tant que premier auteur fut la recherche et le développement de la méthode, les tests de performance ainsi que la rédaction de l'article.

Machine Vision and Applications manuscript No.
(will be inserted by the editor)

Mathieu Marquis-Bolduc · François Deschênes

Occlusion Event Detection using Geometric Features in Spatio-temporal Volumes

Received: date / Accepted: date

Abstract In video sequences, edges in 2D images (frames) produces 3D surfaces in the spatio-temporal volume. In this paper, we propose to consider temporal collisions between edges, and thus objects, as 3D ridges in the spatio-temporal volume. Edges collisions (i.e. ridge points) can be located using the maximum principal curvature and the principal curvature direction. Based on the detected ridges, we propose a technique to identify overlapping objects events in an image sequence, by neither computing depth or optical flow. We present successful experiments on real image sequences.

Mathieu Marquis-Bolduc
Département d'informatique
Université de Sherbrooke
2500 boul. Université
Sherbrooke, Qc, Canada, J1K 2R1
E-mail: Mathieu.Marquis-Bolduc@usherbrooke.ca

François Deschênes
Université du Québec en Outaouais
283, boul. Alexandre-Tache
C.P. 1250, succ. Hull
Gatineau, Qc, Canada, J8X 3X7
Tel.: 1-819-595-3980
Fax: 1-819-595-3825
E-mail: Francois.Deschenes@usherbrooke.ca

Keywords Event detection · Occlusion detection · Occluding edge · Spatio-temporal volume · Spatio-temporal surface

1 Introduction

1.1 Context

The detection of occluding contours and occlusion of objects in an image sequence is a problem of importance. It provides essential clues on the structure and behavior of objects in a sequence. For example, if we can detect that a man is moving in front of a given object, we know that this man is closer to the camera than the object. Our goal is to identify occurrence of occlusion in a digital video sequence without prior knowledge on the scene structure, its lighting, camera movement or intrinsic parameters. This type of semantic information may obviously be useful, for example, in the field of content-based video indexing. The detection of occlusion also has many other applications, including improving scene reconstruction [31] and object tracking [17].

In this paper, we propose a novel method to detect occlusions in a video sequence. More especially, we suggest to consider geometric features of the image sequence in the space-time domain. We consider an image

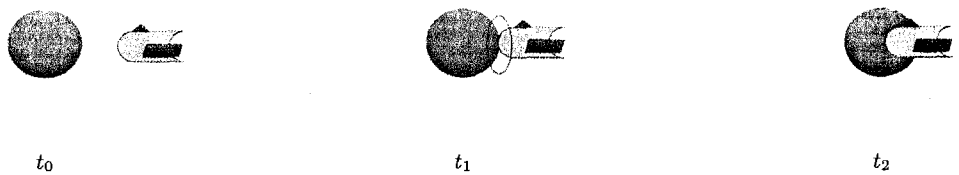


Fig. 1 An occlusion event occurs as one object moves in front of the other

sequence as a videocube [27] and [16] or spatio-temporal volume [28] from which we want to extract the relevant 3D features. This spatio-temporal volume has proven to be useful to detect various types of information [25] and [23] and [14] and [18] and [30]. From this medium, we propose to first identify 3D surfaces formed by edges in the spatio-temporal volume using a 3D gradient edge detector (color or grayscale version). Second, we use the fact that, by extension, collisions of edges in an image should appear as intersections of surfaces in the spatio-temporal volume. We define such a collision of image edges as an *occlusion event* (see Fig. 1). Intersection of surfaces in a volume are expected to form 3D ridges. Those ridges can be located by extracting local maximum curvature, as suggested by Monga [22]. Finally, we use the orientation of colliding surfaces to classify occlusion events as well as identifying the occluding edge. Approaches based on 3D curvature like the one we propose remain uncommon. This method provides extensive information about occlusion events in a video sequence, by neither computing scene depth nor optical flow.

2 Related Work

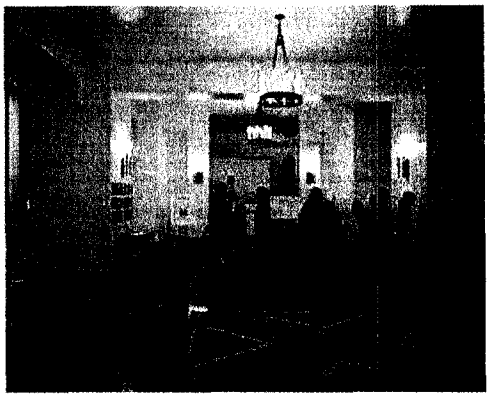
Existing approaches to identify occlusion edges and occlusion events in a video sequence can be grouped in three classes.

The first class of approaches is based on information extracted from a single image. For instance, "T-Junctions" (junctions of image edges forming a T-like structure) may be considered as a cue of occlusion, because they likely appear when an object is partially occluded (see [9]). Texture warpage [11], the deformation of texture caused by object shape, is also an important cue to identify occluding edges of regular, non-planar objects. Finally, changes in object illumination near occlusion edges [15] can also be considered. However, methods based on a single image can return unwanted results in the context of a video sequence. For example, let us consider a scene containing a large poster on a wall. Such methods would identify occluding edges inside the poster, even if those occlusions do not exist *in the scene.* To distinguish such occurrences from real scene occlusions, motion information can be useful.

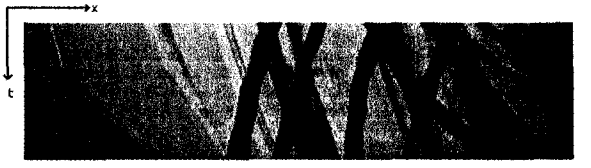
The second class of approaches is based on image matching [6] and [5] and [7]. In these approaches, dense or sparse feature tracking are used to build a model of motion, and occlusion can be detected at motion boundaries. However, without *a priori* knowledge, the computation of image correspondence or optical flow are difficult or unreliable in areas surrounding occluding contours [3]. The optical flow computation can also be affected by changes in the scene lighting. To address such issues, a priori knowledge of occluding edges is of great interest. For instance, it could greatly improve the accuracy of optical flow estimation by avoiding to match occluded regions.

The third class of approaches is based on the study of 3D surfaces in spatio-temporal volumes. Zetsche *et*

al [8] suggest to use the gaussian curvature of points in the spatio-temporal volume to detect occlusions in zones of significant curvature in the XY plane (Image junctions) in the context of optical flow analysis. Konrad and Ristivojevic [18] introduce "object tunnels" that 2D objects of static shapes form in the spatio-temporal volume. By finding the boundaries of spatio-temporal tunnels that become aligned with the temporal axis, they attempt to detect occlusions with at least one static object. The authors mention that their method can however only detect occlusions by static objects with straight line boundaries. Konrad and Ristivojevic also used the object-tunnel model for the segmentation of video sequences and the detection of background occlusions [29]. In both works they attempted to track the tunnel frontiers using the level-set approach. In [1], Mitiche *et al* attempted to detect spatio-temporal surfaces formed by occlusion edges in order to track objects, using a level-set approach that formulates the surfaces as a Bayesian image partitioning problem. As can be seen, most of the methods based on spatio-temporal surfaces work on individual surfaces, and do not consider the interaction between such surfaces that happens at occlusion events. In that regard, some methods (e.g. [2]) aim to find occlusion events by detecting T-Junctions in epipolar planes images (EPI), i.e. slices of the spatio-temporal volumes taken in the epipolar direction. As shown in Fig. 2b, spatio-temporal surfaces form 3D ridges at occlusion events that can be seen as T-junctions if viewed from the appropriate plane in the spatio-temporal volume. Laptev [19] argued that corner-like features, i.e. points that vary into 3 orthogonal directions in the spatio-temporal volume, are a robust sign of significant local, temporal events. He uses a 3D version of the Harris corner detector to detect "local motion events", and shows occlusion events as an example of the features that are detected.



a) Frame sample (t=0)



b) EPI Image

Fig. 2 An EPI image (b) of a video sequence (a) featuring several occlusion events. The EPI image only shows occlusion events that occur in X direction.

Based on this short overview, it is obvious that intersections of spatio-temporal surfaces represent an important cue of occlusion and that their detection is a step further in understanding occlusion phenomenon, that is some occlusion edges identified using spatial (image-level) information can only be confirmed or rejected by also taking into account dynamic (temporal) information.

Our contribution can be seen as a generalisation of Apostoloff and Fitzgibbon's work [2], since we generalize the concept of T-Junctions in the EPI (see Fig. 2) to the broader concept of ridges in the spatio-temporal volume. It can also be seen as a continuity of Laptev's [19] work, as from all significant local, temporal events we deduce significant properties of the occlusion/dis-occlusion event. We provide a thorough analysis of this specific event, as well as means of discriminating it from other "local motion events".

In the next section, we present a definition of edge collisions occurring at occlusion events using 3D ridges. We then explain how ridges are located using 3D curvature information. Following this, we will discuss how to identify relevant ridges. Finally, we present an application that uses both the ridge and spatio-temporal surface information to distinguish occlusion from disocclusion events and detect which object overlaps another one in a video sequence.

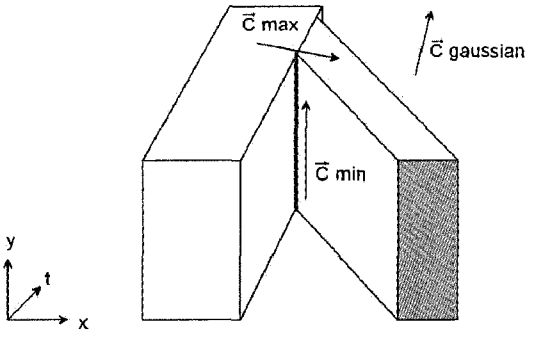


Fig. 3 Two rectangles move toward each other. As their sides collide, a ridge is formed (in the middle). Here, the left rectangle occludes the right one by moving over it.

3 Detecting occlusion events

3.1 Analysis of occlusion events in the spatio-temporal volume

As a given 2D point moves from image to image in a video sequence, it forms a curve called a spatio-temporal (ST) curve. In the same manner, curves in 2D images form spatio-temporal surfaces, and areas form spatio-temporal volumes, sometimes called tunnels [18]. The tracking of spatio-temporal curves, surfaces or volumes is usually concentrated on pixels that are part of an edge in 2D images due to the aperture problem. By extension, we will only consider pixels that are part of 3D surfaces that edges form in the space-time domain. We detect those surfaces by applying a 3D gradient operator on the spatio-temporal volume. As will be seen, the use of the 3D color gradient of the spatio-temporal volume allows to convert the input data (video sequence) into a form that makes spatio-temporal surfaces explicit, which we call the *gradient volume.*

Spatio-temporal surfaces interacting together usually create regions of high 3D curvature in the spatio-temporal volume. It has been noted before that lines found in time slices intersect each other (e.g. T-Junctions) when occlusion occurs [23] and [34] and [2]. However, we would have to take an infinity of slices to be able to detect all the different angles and speed at which those corners can appear. Taking a slice in the EPI allows to detect occlusion events that occur along the main motion direction, but several important motions can be present in a scene. Since those intersecting lines correspond to slices of spatio-temporal surfaces, related corners (or T-Junctions) are in fact slices of a ridge in the spatio-temporal volume, formed by the intersection of two surfaces. (see Fig. 3).

A critical observation at this point is that, for semi-rigid objects, *at least one of the two surfaces corresponds to an occluding contour* in the image sequence; both surfaces cannot correspond to texture edges. In the later case, it is obvious that they will not meet, or one edge would have been occluded before the collision could occur. Also note that in the presence of rotation, a texture edge may become an occlusion edge.

As mentionned before, ridges are primarily points of high 3D curvature. The structure of ridges caused by colliding edges can then be defined using 3 different 3D curvature directions $\vec{C}_{max}$, $\vec{C}_{min}$ and $\vec{C}_{gaussian}$ [10] (see

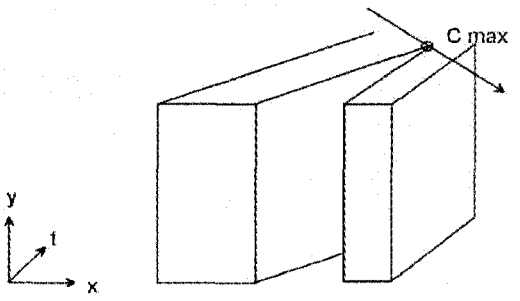


Fig. 4 The principal curvature direction of two objects moving in the same direction but at different speeds

Fig. 3):

Principal curvature direction

The principal curvature direction $\overrightarrow{C}_{max}$ of any ridge is the direction in which the magnitude of the curvature is the highest. On occlusion event ridges, it is "across" the ridge, usually perpendicular to the occlusion edge. An important observation is that the temporal component of the normalized principal curvature direction vector depends on the relative velocity of the objects. If both edges at the occlusion events are moving at the same speed toward each other in the video sequence, then the temporal component of the principal curvature direction will be null and the direction will lie on the image plane exclusively. As the sum of the relative speed of the objects increase, so will the temporal component. Consider, for example, two objects moving in the same direction, one faster than the other (see Fig. 4). The temporal component of $\overrightarrow{C}_{max}$ will then be large. However, the principal curvature direction cannot have an exclusive temporal component (i.e. $\overrightarrow{C}_{max} = (0, 0, 1)$), otherwise the colliding spatio-temporal surfaces would be confounded. This first direction, along with the curvature value, are the features we will principally focus on, since their computation are the most reliable (see section 3.2).

Secondary curvature direction

The next feature is the secondary curvature direction $\overrightarrow{C}_{min}$, the direction in which the curvature magnitude is the lowest. For collision ridges, it is "along" the ridge, and depends on the orientation (in the image) of the surfaces that are colliding. In simpler terms, this direction roughly corresponds to the orientation of the image edges as they collide at the occlusion event. Obviously, this direction should always lie in the XY image plane.

Gaussian curvature direction

Finally, there is the gaussian curvature direction $\overrightarrow{C}_{gaussian}$, defined as the cross-product of the two previous ones. Due to the constraint on the principal and secondary curvature directions, the gaussian curvature direction never lies in the XY plane. Notice that we do not necessarily use this direction in our analysis since experiments confirmed that its computation is not reliable in some cases, such as when the magnitude of one or both of the previous curvature directions is close to zero.

Based on this description of occlusion ridges, we observe that the sharpness (magnitude of curvature) of a ridge should depend on the relative speed of objects. Collision of edges moving quickly toward each other will cause flat ridges with a low principal curvature. Collisions between spatio-temporal surfaces with a normal vector close to or on the XY plane will cause sharp ridges with a high principal curvature. At any rate, points on an occlusion ridge should locally have higher curvature value than points on spatio-temporal surfaces not involved in a collision at that time. Thus, according to our analysis, the first step for identifying occlusion events is to detect 3D ridges caused by collisions of surfaces by computing and then thresholding 3D curvature of spatio-

temporal surfaces. Related orientation information can then be useful for discriminating and classifying those 3D ridges.

3.2 Detection of 3D Ridges

Detection of 3D ridges is difficult because computation of reliable 3D curvature information is a complex matter [13]. Most of the work in this area has been concentrated on the use of triangular meshes or similar data (e.g. [24] and [33] and [21]) or height-function / range sensor data (e.g. [20]), and are not appropriate to use in dense, volumetric data. An interesting development is the emergence of methods that do not require computation of partial derivatives, such as tensor voting schemes [32]. However, to the authors' knowledge, the only existing methods that can be directly applied to compact volumetric data are based on partial derivatives of the volume. For this reason, we base our method to compute 3D curvature information at image edges location on a method first devised by Monga [22] to extract crest lines in medical volumetric data. This type of data is not too different from our own spatio-temporal data in term of density, representation and presence of noise. This method also provides all the necessary information about principal curvatures along with their directions. Let us review it briefly.

This method uses the partial derivatives I of the 3D data to compute the principal curvature of the hypersurface (in our case, the spatio-temporal volume) as a valid approximation of the principal curvature of the surface. First, the first and second order partial derivatives of the volume need to be computed. Then, the first (Eq. 1) and second (Eq. 2) fundamental forms of the hypersurface are computed:

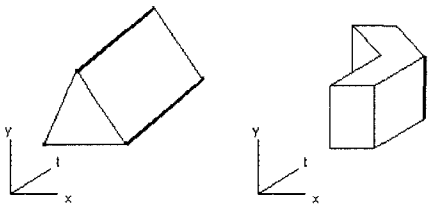
$$F_1 = \begin{bmatrix} 1+I_x^2 & I_xI_y & I_xI_z \\ I_xI_y & 1+I_y^2 & I_yI_z \\ I_xI_z & I_yI_z & 1+I_z^2 \end{bmatrix}, \tag{1}$$

$$F_2 = \frac{1}{\sqrt{1+I_x^2+I_z^2+I_y^2}} \begin{bmatrix} -I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & -I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & -I_{zz} \end{bmatrix}. \tag{2}$$

Using the fundamental forms we then compute the Weingarten endomorphism:

$$W^t = F_2 F_1^{-1}. \tag{3}$$

Finally, the endomorphism's eigenvalues and eigenvectors are computed. To this end, several algorithms can be used (e.g. [26] and [4]). Note that since this matrix is generally not symmetric, there is no way to ensure that there will be more than one real eigenvalue. For this reason, ridges are located by only considering the principal curvature of the hypersurface, that is the real eigenvalue having the highest magnitude, and the corresponding eigenvector as direction: $C = max(|\lambda_i|), i = 1, 2, 3$ and λ_i are the real eigenvalues of W^t. If available, the second and gaussian curvatures are used to refine the location and also during the classification process (see sections 3.3 and 4). Ridge (or crest line) points are defined as points having maximum curvature magnitude in the principal curvature direction. By applying this method to the 3D gradient of the spatio-temporal volume, we directly obtain ridge points of the spatio-temporal surfaces. Before being able to classify occlusion (and "discocclusion" events), we now need to classify those ridge points to keep only those that represent collisions between spatio-temporal surfaces at these events.



a b

Fig. 5 Example of non-occlusion ridges caused by the motion of simple forms. (a): The corners of the triangle form ridges over the temporal sequence. (b): A sudden change of speed and direction can also cause a ridge in the spatio-temporal volume.

3.3 Discriminating 3D Ridges

The method briefly presented in previous sections computes all of the ridges of 3D surfaces. However, in practice, not all of the ridges located by this technique originate from edge collisions. This was also noted by Laptev 's work on spatio-temporal events [19]. We must hence filter out the non-relevant ridges.

Ridges due to noise and numerical imprecision

Both noise and numerical imprecision can cause false positives. To circumvent this problem, we keep only ridges that have coherent curvature. We filter out ridges with aberrant curvature (so high or so low that we are not able to detect them). The value of the lowest threshold depends on the size of the support used to compute the derivatives. Objects that are moving too fast will not be inside the support in adjacent frame even if they are involved in a collision. A ridge that is too sharp, representing a collision that happens between objects that are moving slowly is also a degenerate case due to spatial sampling. According to the definition of curvature, it can be shown that the range of the curvature values expected is approximately $[\frac{1}{W}, 1]$, where W is the half-size of the image support in pixel units.

Ridges due to 2D corners

As seen in Fig. 5, 2D corners in images will also create ridges in the spatio-temporal volume, and they are expected to have principal curvatures in the same range as our collision ridges. Their principal curvature direction $\vec{C}_{max}$ should always lay close to the XY plane. Unfortunately, some collision ridges, for example two edges moving directly toward each other at the same speed, are also expected to have a principal curvature direction that lies close to the XY plane. So the principal curvature directions cannot be used to discriminate them. They however do present a different secondary curvature direction. Ridges caused by occlusion events should have a secondary direction close to the XY plane, that is with a low temporal component, while corner ridges should have a high temporal component. Thus, theoretically, the secondary curvature direction could be used to discriminate such ridges. Also, the orientation of corner ridges will make them appear in each frame as single isolated points or short broken lines, while collision ridges appear as long and continuous lines or curves at collision time. So the size of the event in each frame could also be used to discriminate them.

Unfortunately, the secondary direction cannot always be reliably computed with the method presented in Section 3.2. However, experiments have confirmed that corner ridges can be considered as noise in each frame since they are isolated and can be filtered out using standard noise reduction techniques.

Ridges due to sudden changes of velocity

Finally, we consider ridges caused by a sudden change of speed or direction (Fig. 5b), that can be rightfully con-

sidered as significant spatio-temporal events [19]. In natural image sequences with sufficient temporal sampling, the curvature of those ridges is usually quite low and they can thus be filtered out by applying temporal smoothing and lower curvature thresholding. Notice that the lack of a second spatio-temporal surface in the vicinity of those ridges is another cue to filter out false collisions. Thus it will also be possible to discriminate them using additional (non-curvature) information in the classification step.

4 Classification of occlusion events

We now show how ridges extracted using the method described in Sections 3.2 and 3.3 can be applied to event detection. We want to detect objects that move in front of or behind other objects, and possibly discriminate them from other events that yield a similar spatio-temporal ridge, including the previously mentioned sharp changes of velocity, and also moving objects which edges come into contact but without overlapping. Using occlusion event detection, we can detect not only when two image edges collide (or "discollide", in the case of a disocclusion event), but also the orientation of this collision. This information allows us to develop a simple model for detecting overlapping objects. We know that in the case of an occlusion event, at least one of the implicated edges is an occlusion edge, that is the physical border of an object. We now need to detect which object, if any, moves over the other one.

Occlusion planes

Using the different curvature directions, we can describe three planes passing by a ridge point that will be useful for describing occlusion events. These planes and their properties will be useful in our classification of occlusion events. We define the *occlusion time plane* as the plane with normal (0,0,1) that is located at the time of occlusion. This plane has the obvious characteristic that every edge points on one side of the plane is part of a spatio-temporal surface *before* the occlusion event, and every edge point on the other side of the plane is part of a spatio-temporal surface *after* the occlusion event. The second plane is the *occlusion edge plane*. This plane's normal is lying on the image (XY) plane and is perpendicular to the secondary curvature direction of the ridge. This plane passes by the occlusion point and thus has the characteristic of separating two surfaces that intersect at an occlusion event. The last one, the *occlusion ridge plane*, is perpendicular to the first two. If we recall spatio-temporal slices methods, this plane would be the ideal slice to use in order to observe its associated occlusion event.

The surface orientation model

By considering the previous definitions, we know that an important cue is the presence of one spatio-temporal surface on one side of the occlusion time plane, and of two (or more) on the other side. If we can identify to which object belongs the singular surface, then we will not only be able to classify the event as occlusion or disocclusion, but we will also have identified a true occlusion edge. To this end, we have to consider what information is reliable about the spatio-temporal surfaces. Obviously, the "magnitude of the surface", that is the magnitude of the 3D gradient in the spatio-temporal volume, is going to vary significantly along any spatio-temporal surface. This not only makes its direct use difficult, it also makes the computation of the *spatio-temporal surface area* correspondingly problematic, which has been confirmed by our experiments. The next information about the spatio-temporal surfaces surrounding the occlusion event that we can make use of is the orientation. The orientation

of the surfaces can be computed using derivative filters on the gradient volume. According to our model, on one side of the occlusion time plane we should find only the main class of orientation, while we may find two or more classes on the other side. Projecting the spatio-temporal orientation on the occlusion edge plane allows us to simplify the classification of occlusion events. Let us denote the surface normal at any given point as $\overrightarrow{\theta}$. Remember to normalize all $\overrightarrow{\theta}$ such that the surface normals are always facing in the same temporal direction. The relative orientation of a surface point with respect to the occlusion edge plane can be expressed as:

$$\theta = \cos^{-1}(\overrightarrow{\theta} \bullet \overrightarrow{C}_{min}). \tag{4}$$

where $\bullet$ is the standard dot-product. To identify on which side of the occlusion time plane the surface orientation vary the most, we compute within a given neighborhood both the mean and variance of θ: $\mu(\theta)$ and $\sigma^2(\theta)$. We do so for two equally sized half-sphere neighborhoods before $(t-)$ and after $(t+)$ the occlusion time plane, that is: $\mu(\theta_{t-}), \sigma^2(\theta_{t-}), \mu(\theta_{t+}), \sigma^2(\theta_{t+})$. We then propose to define the following occlusion metric:

$$\Delta = \sigma^2(\theta_{t-}) - \sigma^2(\theta_{t+}). \tag{5}$$

With this metric, classifying the occlusion event is easy. For some positive threshold Δ_{min} If $\Delta > \Delta_{min}$, then the occlusion event is an *occlusion*. If $\Delta < -\Delta_{min}$, then the occlusion event is a *disocclusion*. Otherwise, in cases where $|\Delta| < \Delta_{min}$, there is no certitude of occlusion/disocclusion despite the ridge. It could be noise, transparent occlusion (e.g. occlusion by a shadow) or two objects that hit together without overlapping. In cases where we do find occlusion or disocclusion, the relative orientation $\mu(\theta)$ corresponding to the lower $\sigma^2(\theta)$ can be used to identify the occlusion edge.

To identify reliable orientation information, some precautions must be taken. First, surface points too close to the occlusion event must not be taken into account, because their orientation are likely to be influenced by the event. To this end, we take only points in the neighborhood that are sufficiently far from it. If (R_x, R_y, R_t) is the position of the occlusion event under classification, then we take only surface points (x, y, z) where:

$$R^2_{max} < (x - R_x, y - R_y, t - R_t)^2 < R^2_{min}. \tag{6}$$

where R_{min} is the minimum distance threshold from the occlusion point and R_{max} is the maximum distance threshold from the occlusion point. Second, we only consider points for which the magnitude of the orientation vector is significant when compared to the gradient volume value:

$$|\overrightarrow{\theta}| > k * I_G(x, y, t).es \tag{7}$$

where $k \in]0, 1[$ and $I_G(x, y, t)$ is the gradient volume. A useful characteristic of using surface orientation for classification is that, even if "outsider" surfaces are present in the occlusion event neighborhood, they are likely to have an orientation similar to the one involved in the occlusion event, and thus are unlikely to influence results in a wrong direction. One can observe this effect in Fig. 6 (d). Both this characteristic, and the usefulness of the classification method are confirmed by our experimental results.

Overview of the algorithm for occlusion event detection and classification

Before we present our results for detection and classification of occlusion events in the spatio-temporal volume, let us review the complete algorithm.

Input:

- Video sequence filmed in a single shot.

Output:

- Location of occlusion events

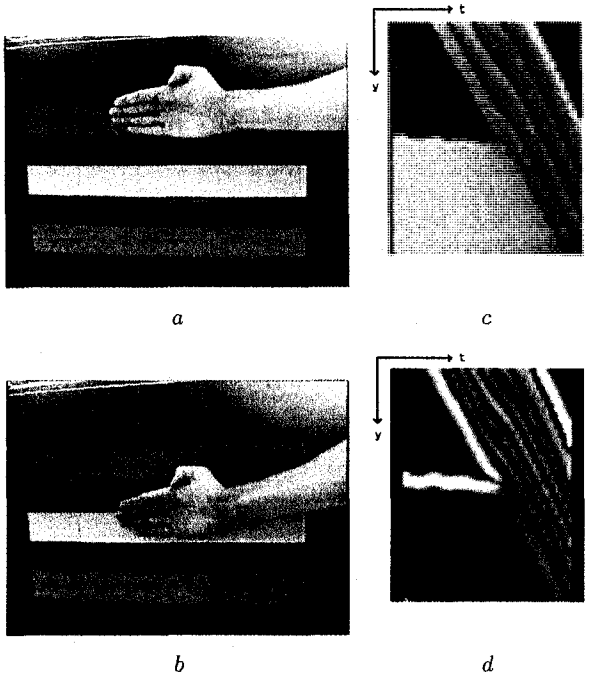


Fig. 6 Closeup view of an occlusion event as the hand pass over the white band a) The video sequence at t=0. b) The video sequence at t=50. c) A close-up view of the occlusion event in the spatio-temporal volume, taken from the occlusion ridge plane. d) The same view in the gradient volume. Spatio-temporal surfaces can be clearly observed.

- Type of occlusion for each event (i.e. occlusion or disocclusion)
- Occlusion edge location

Parameters:

- Size of the derivative filters
- Temporal derivative threshold
- Curvature magnitude thresholds
- Principal curvature direction threshold
- Size of the classification neighborhood
- Surface orientation threshold
- Gradient orientation threshold

Main steps:

1. Computation of principal curvature and related directions
   - *a*– Compute 3D gradient volume of the spatio-temporal volume (color or grayscale);
   - *b*– Compute first and second order derivatives of the gradient volume;
   - *c*– Compute principal curvatures and directions of the volume using equations (1), (2) and (3).
2. Locate occlusion events
   - *d*– Find and threshold the principal curvature maxima in the principal curvature direction
   - *e*– Filter out false positives using principal curvature directions (cf. Section 3.3).
3. Classify occlusion types
   - *f*– For a neighborhood on one side of the occlusion time plane and another one on the other side, compute the mean and variance of the spatio-temporal surface orientation relative to the occlusion edge plane;
   - *g*– Classify events as an occlusion, disocclusion or non-occlusion using the orientation variances;
   - *h*– Identify occluding edges using the mean surface orientation corresponding to the lowest variance.

5 Performance evaluation

5.1 Performance of occlusion event detection

In this section, we evaluate the performance of the proposed method for the detection of occlusion events. We also evaluate the influence of the parameters for the proposed method. The first set of 24 sequences used in our evaluation were filmed using a Sony hand camera. These sequences all feature events that can be labelled by hand, and cover all different situations in regard to occlusion events, e.g.:

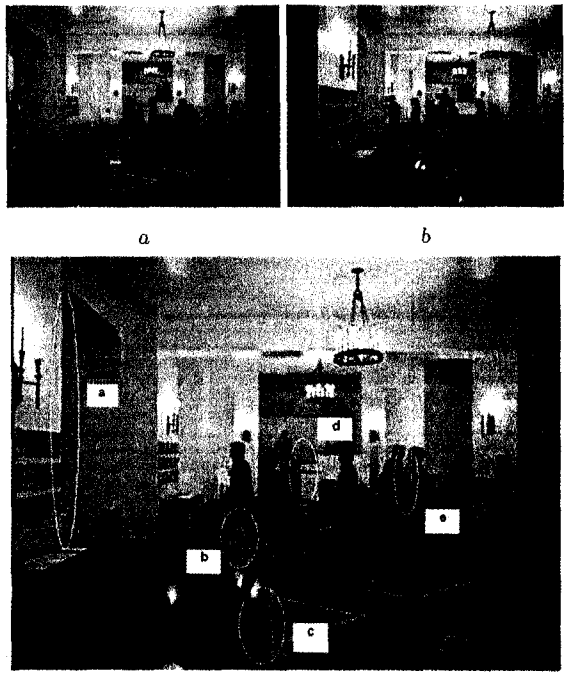
- occlusion vs disocclusion
- stationary vs moving objects

- objects with irregular occlusion edges
- strongly textured objects
- changes of velocity
- transparent objects
- etc.

An additional set of 5 sequences were extracted from existing movies to provide additional variety. Sequences are made of frames containing 320x200 pixels at 15 frames per second. We used gaussian derivative filters of scale σ to compute the partial derivatives involved. This parameter was tested with values ranging from 1.2 to 1.8.

Fig. 8 shows an example of occlusion event detection on a simple sequence. As can be seen, the occlusion between the two objects is correctly identified, and the detection signal covers almost the entire occlusion edge. Fig. 7-(c) shows a subset of the wide variety of occlusion events that can be detected at this step, including: (a)-Occlusion between walls caused by camera movement, (b)-Repeated occlusion and disocclusion of the legs of a walking person, (c)-occlusion between a person's shadow and the carpet, (d)- dis-occlusion as a background object re-appear from behind a walking person, (e)-occlusion of a moving person by another.

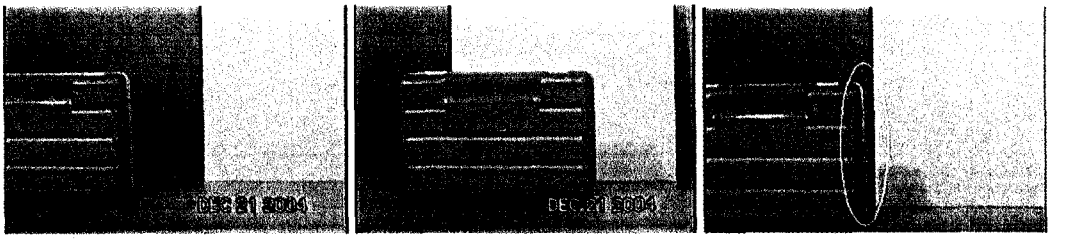
The results of all of the experiments are summarized in Table 1. The Detection Rate represents the proportion between the size of the detected occlusion events and the size of the actual events in the video frames. The False Detection Rate represent the proportion between the size of the error signal and the size of the correct signal. For example, if a particular occlusion event is 20 pixels long in the occuring video frame, and we detect a total signal of 18 pixels, 15 of which are in the correct occlusion event location and 3 of them being noise. The Detection Rate would be $15/20 = 75\%$, while the False Detection Rate would be $3/15 = 20\%$. Ground truth was estimated visually. Overall, the performance of the proposed method was satisfactory, showing that the proposed method could detect occlusion events with sufficient accuracy, and can discriminate them from other significant spatio-temporal events. Note that further false positives may still be eliminated in the classification process.



c

Fig. 7 A complex video sequence featuring dozens of occlusion events. Only a subset of the events are shown, for visibility purpose. a) The sequence at frame t = 0. b) The sequence at frame t = 49 c) The sequence at frame t = 22, showing examples of occlusion events detected in the sequence

While occlusion events with a high relative object velocity can only be detected with a large filter (and a large smoothing value in our filters's case), the proposed method performs better when velocity is between 1 and 5 pixels per frame, high velocity events tend to have undesired side-effects, as confirmed experimentally. For example, if the object displacement per frame is larger than the object themselves, self-occlusions will be consistently



a *b* *c*

Fig. 8 In this test sequences, both objects are moving toward each other. (a) The sequence at frame t = 0. (b) The sequence at frame t = 50 (c) The occlusion event detected at t = 22.

Table 1 Occlusion events detection results for all video sequences, with occlusion velocity ranging from 0.3 to 18 pixels / frame.

σ	Mean Detection Rate	Detection Rate Standard Deviation	Mean False Detection Rate
1.2	0.75	0.167	0.15
1.4	0.783	0.089	0.14
1.6	0.82	0.127	0.17
1.8	0.75	0.151	0.21

Table 2 Occlusion events detection results for video sequences with relative occlusion velocity ranging between 1 and 5 pixels per frame.

σ	Mean Detection Rate	Detection Rate Standard Deviation	Mean False Detection Rate
1.2	0.875	0.075	0.12
1.4	0.85	0.073	0.13
1.6	0.875	0.078	0.17
1.8	0.8	0.15	0.19

detected for those objects. This is also true if a moving object reverse its direction in a very abrupt manner in sequences with insufficient temporal sampling. This is confirmed by Table 2. This table presents results obtained with a subset of video sequences containing objects that are moving from 1 to 5 pixels per frame. As can be seen, detection rate is higher while false detection rate is lower.

In a different line of thoughts, from Tables 1 and 2, we observe that the algorithm performs well with σ ranging from 1.2 to 1.6. As can be clearly seen from Table 2, the scale parameter influences the location of occlusion events. This leads to an increasing false detection rate as σ increase. This effect is not a surprise as it is consistent with the proposed occlusion ridge model. Moreover, this effect related to the scale is well known in 2D corner detectors [12]. As expected, large σ values also have the expected effect of lowering the primary curvature value of the detected events.

In addition to this parameter, we used several others to discriminate events based on the principal curvature value, the temporal (t) element of the principal curvature direction, the temporal partial derivative (dz) on the original image sequence and finally the size of the events themselves. Principal curvatures values of occlusion events are usually in a range of $[0.05, 1.1]$, as expected with the proposed model. However, we expected these values to correlate with the velocity of the collision, but our tests show that this is not consistent. We can only conclude that some other factors have an effect on the principal curvature, such as the colliding spatio-temporal surfaces individual magnitude and direction. The principal curvature direction proved to be the most

helpful factor to discriminate occlusion events from other points of local maximum curvature. The maximum magnitude of the temporal component of its normalized vector was below 0.6 in all but one of our test. Thresholding this value to 0.6 eliminate most of the undesired events. After thresholding both the previous values in the indicated ranges, several events were still present, generally in areas of high 2D curvature in the image plane. Most of these were eliminated by keeping only events where the temporal partial derivative (dz) on the original image sequence is at least 20 (on a 8bit color resolution), and by keeping only events that are 10 pixels-wide in each frame.

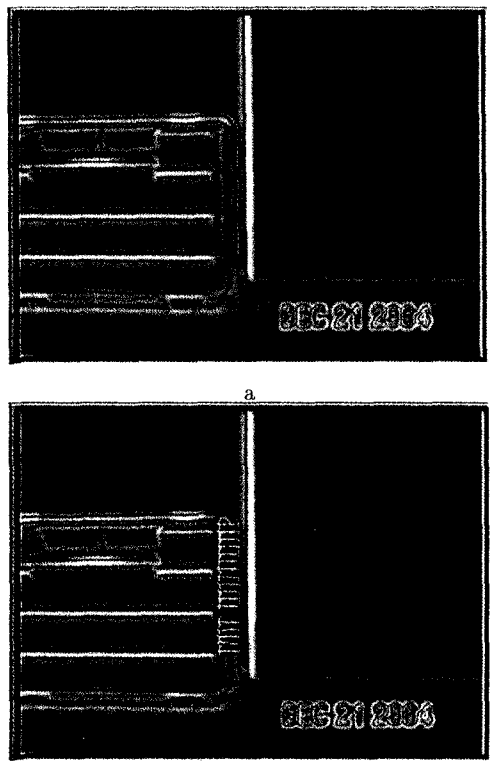
5.2 Performance of occlusion event classification

We now present results of the occlusion event classification of all of the events that were detected above. Fig. 9 shows the result of the event classification of the example presented in Fig. 8. As can be seen, most of the detected occlusion signal has been correctly confirmed as being an occlusion, rather than a dis-occlusion or a false positive. No part of it has been mis-labelled as a dis-occlusion. It can also be seen that the occluding object has been correctly identified. Results of occlusion even classification using the proposed method are summarized in Table 3. The correct type of occlusion event was identified with a success rate of 94-95%, and the occlusion edge was correctly identified in all cases. Fig. 10 shows an example of a simple occlusion event viewed on the occlusion ridge plane in the gradient volume. It also shows the neigborhood used in the classification process. Fig. 11 shows the surfaces orientation variability for the same event. As you can see, the variability before and after the occlusion event is very distinct, and allows us to classify this event as an occlusion rather than a dis-occlusion. We used a Δ_{min} threshold of 1.0 in order to classify our events. This also allowed a reduction of the noise in our result.

Neighborhoods of a radius $R_{max} = 15$ gave excellent results for all well-sampled events where the relative objects velocity was within acceptable values (between 0.5 and 5 pixels per frame), with $R_{min} = 0.5 * R_{max}$. Note that a significant amount of noise was eliminated in the classification process. For events where the velocity is higher than 5 pixels per frame, R_{max} must be lowered accordingly. In such under-sampled video sequences, spatio-temporal surfaces orientation may change faster, and occlusion events are more likely to happen closer to each other, so a smaller neighborhood must be used to reduce noise. For our fastest event, with a relative velocity of 18 pixels per frame, simply lowering R_{max} to 12 gave optimal results. The classification process did not succeed for over-sampled video sequences where the velocity is lower than 0.5 pixel per frame, regardless of the size of the neighborhood. This is because the spatio-temporal surfaces involved in the events stay too close to each other for their individual orientation to be reliably computed, while increasing R_{max} add too much noise in the neighborhood. Still, it is possible to use the proposed occlusion event classification method on those video sequences by simply re-sampling the video to a more adequate time interval.

5.3 Influence of noise on occlusion event detection

We tested the influence of noise on the detection of occlusion events. For this purpose we added Gaussian noise our set of sequences. We vary the standard deviation of the noise (σ_{noise}) from 0 to 30 on a 0-255 scale. Our re-



a

b

Fig. 9 The spatio-temporal volume from the sequence shown in picture 8a-c, at frame t=22. a) Points confirmed as occlusion events (in red) in the classification step. b) The occluding object has been correctly identified (as shown in yellow arrows).

Table 3 Occlusion events classification results for all video sequences with relative event velocity between 0.5 and 18 pixels per frame

Velocity Range (pixels /frame)	Neighborhood radius used	Correct Classification Rate	Occlusion Edge Identification Rate	Residual False Positive
[0.5-5]	15	0.94	1.0	0.08
]5-18]	12	0.95	1.0	0.09

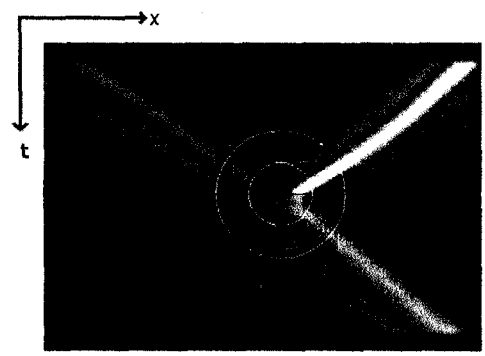


Fig. 10 A cross view of the classification of the collision event presented in Fig. 8a-c. Closer than R_{min} (the smallest circle), it is difficult to make up a surface orientation. Between R_{min} and R_{max}, we can observe two distinct surface orientations before the occlusion (t_-) and only one surface orientation after the occlusion event (t_+). The horizontal line (red) represent the occlusion time plane.

sults, shown in Fig. 12 demonstrate that noise has very little effect on the effectiveness of the proposed method, with the detection rate and false detection rate remaining unchanged. This can be explained first since Gaussian noise, having no spatial or temporal cohesion, has very little effect on the structure of the spatio-temporal volume. Also, the Gaussian derivative filters used in steps a and b of the proposed method also act has denoising filters. An example of comparative result with a noisy sequence can be seen in Fig. 13. This figure clearly shows that occlusion event detection can accurately be detection even with severe noise.

Overall, all of the previous results confirm the accuracy of the proposed method.

6 Conclusions

We have provided a detailed analysis of the spatio-temporal geometry of collisions between object contours, and how they can be modeled by a 3D ridge in the spatio-temporal

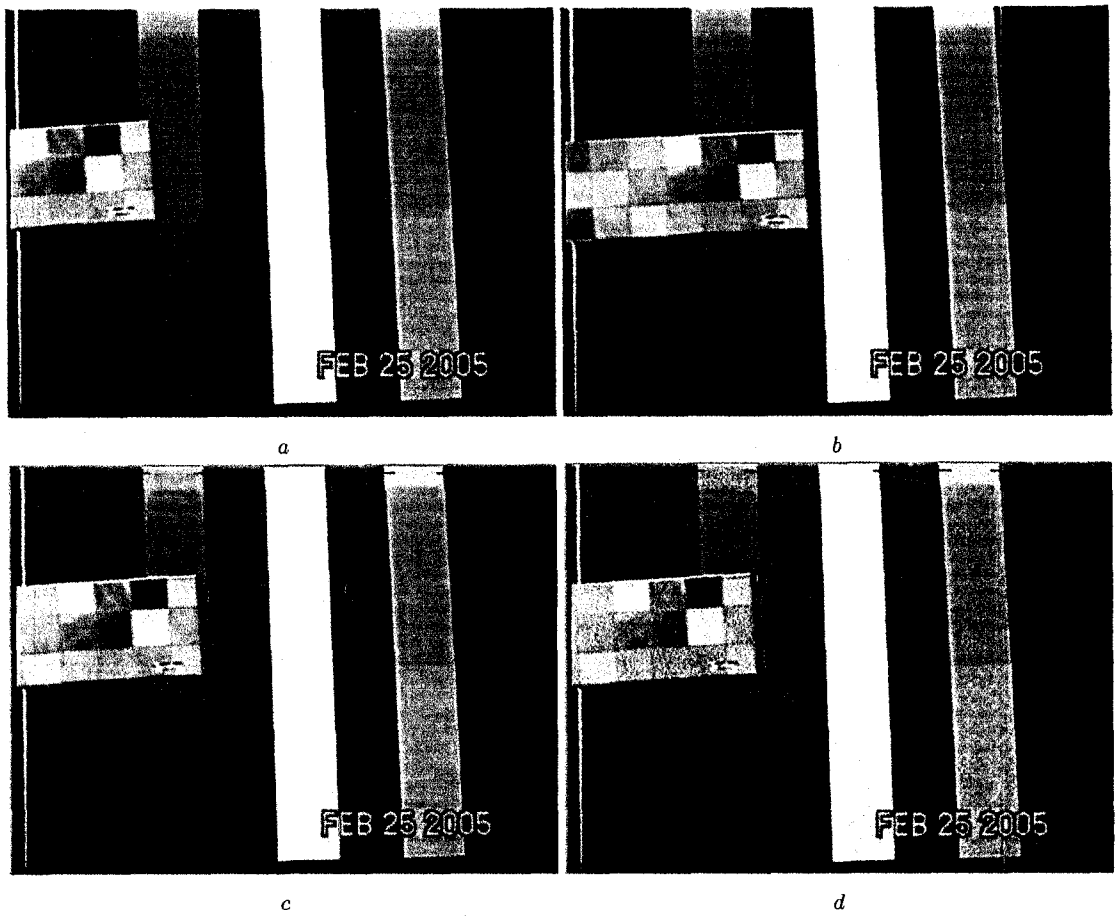


Fig. 13 This test sequence features a textured box moving over color stripes. (a) The sequence at frame t = 0. (b) The sequence at frame t = 24. (c) The original result of occlusion event detection at frame f = 10. (d) The result of occlusion event detection with severe Gaussian noise ($\sigma = 20$) added to the sequence.

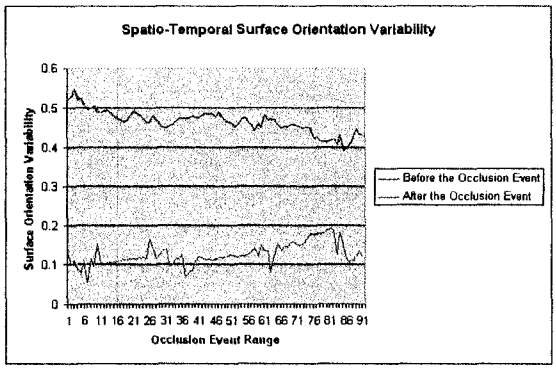


Fig. 11 Spatio-temporal surface orientation variability for the occlusion event presented in Fig. 8(a)-(c).

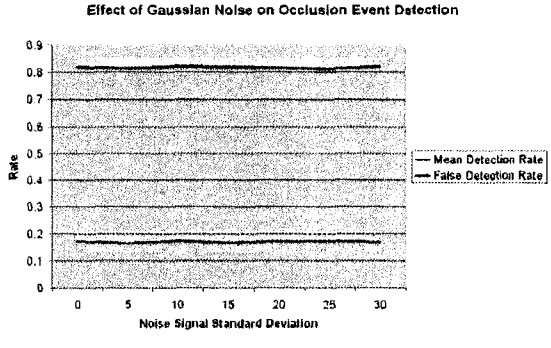


Fig. 12 Effect of gaussian noise on occlusion events detection rate and false detection rate, with noise standard deviation varying from 0 to 30.

volume using curvature information. We then proposed a novel approach to detect such occlusion events. We have done so by neither tracking nor computing scene depth or optical flow. For this purpose, we consider the 3D principal curvatures and the associated directions of the gradient spatio-temporal volume for the detection process. It hence allows us to detect collisions between occluding contours that can be straight, parallel, curved, etc. From this information, we then showed how the addition of spatio-temporal surfaces orientation could be used to classify occlusion events as occlusion or disocclusion, as well as to identify occlusion edges in video sequences. Finally, we presented experimental results on real image sequences. These results confirmed the accuracy of the proposed technique.

Acknowledgements The authors would like to thank Professor Djemel Ziou and Miss Wei Pan for their valuable comments, as well as Mr. Charles Perreault and Mr. Ian Bailey for their contribution with creating and managing the video data. This work is partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) research funds and the Fond Québécois de la Recherche sur la Nature et les Technologies (FQRNT).

References

1. A. Mitiche, R.F., Mansouri, A.: Motion tracking as spatio-temporal motion boundary detection. Robot Autonomous Systems **43(1)**, 39–50 (2003)
2. Apostoloff, N., Fitzgibbon, A.: Learning spatiotemporal t-junctions for occlusion detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR05), vol. 2, pp. 553–559. San Diego, CA, USA (2005)
3. Baker, H., Bolles, R., Woodfill, J.: Realtime stereo and motion integration for navigation. In: Proc. Image Understanding Workshop, pp. 1295–1304 (1994)
4. Bass, J.: Cours de mathematique, 5th edn. Paris : Masson (1977-1978)
5. Black, M., Fleet, D.: Probabilistic detection and tracking of motion boundaries. International Journal of Computer Vision **38**(3), 231–245 (2000)
6. Bleyer, M., Gelautz, M.: A layered stereo matching algorithm using image segmentation andglobal visibility constraints. In: Photogrammetry and Remote Sensing, 59, pp. 128–150 (2005)
7. Bobick, A.F., Intille, S.S.: Large occlusion stereo. International Journal of Computer Vision **33**(3), 181–200 (1999). DOI http://dx.doi.org/10.1023/A:1008150329890
8. C. Zetzche, E.B., Berkmann, J.: Spatiotemporal curvature measures for flow field analysis. pp. 337–350 (1991)
9. Caselles, V., Coll, B., Morel, J.: A Kanizsa programme. Universit Paris-Dauphine (1995)
10. Casey, J.: Exploring Curvature. Wiedbaden, Germany: Vieweg (1996)
11. Clerc, M., Mallat, S.: The texture gradient equation for recovering shape from texture. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, pp. 536–549 (2002)
12. Deriche, R., Giraudon, G.: Accurate corner detection: An analytical study. In: Proc. 3rd International Conference on Computer Vision, pp. 66–70 (1990)
13. Flynn, P., Jain, A.: On reliable curvature estimation. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 110–116. San Diego, California, USA (1989)
14. H. Kawasaki, K.I., Sakauchi, M.: Spatio-temporal analysis of omni image. In: Computer Vision and Pattern Recognition, pp. II: 577–584 (2000)
15. Huggins, P., Chen, H., Belhumeur, P., Zucker, S.: Finding folds: On the appearance and identification of occlusion. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, p. 718. Hawaii, USA (2001)
16. J. Woodring, C.W., Shen, H.: High dimensional direct rendering of time-varying volumetric data. Visualization pp. 414–417 (2003). URL citeseer.ist.psu.edu/woodring03high.html
17. Jepson, A., Fleet, D., Maraghi, T.: Robust online appearance models for visual tracking. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 415–422. Hawaii, USA (2001)

18. Konrad, J., Ristivojevic, M.: Video segmentation and occlusion detection over multiple frames. In: Image and Video Communications and Processing, pp. 377–388. Santa Clara, U.S.A (2003)
19. Laptev, I.: Local Spatio-Temporal Image Features for Motion Interpretation. KTH Numerical Analysis and Computer Science (2004)
20. Lindeberg, T.: Edge detection and ridge detection with automatic scale selection. International Journal of Computer Vision **30**(2), 117–154 (1998)
21. Ma, K.L., Interrante, V.: Extracting feature lines from 3d unstructured grids. In: IEEE Visualization, pp. 285–292 (1997)
22. Monga, O., Benayoun, S.: Using partial derivatives of 3d images to extract typical surface features. Computer Vision and Image Understanding **61**(2), 171–189 (1995). DOI http://dx.doi.org/10.1006/cviu.1995.1014
23. Niyogi, S.A.: Detecting kinetic occlusion. In: International Conference on Computer Vision, pp. 1044–1049. Cambridge, Massachusetts, USA (1995). URL citeseer.ist.psu.edu/niyogi95detecting.html
24. Page, D., Koschan, A., Sun, Y., Paik, J., Abidi, M.: Robust crease detection and curvature estimation of piecewise smooth surfaces from triangle mesh approximations using normal voting. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, p. 162 (2001)
25. Peng, S.L., Medioni, G.: Spatio-temporal analysis of an image sequence with occlusion. In: Image Understanding Workshop, pp. 433–442. Boston, Massachusetts, USA (1988)
26. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical Recipes in C++: The Art of Scientific Computing 2nd edition. Cambridge University Press (2002)
27. R. Rodrigues A.R. Fernandes, K.v.O., Ernst, F.: Reconstructing depth from spatiotemporal curves. In: Vision Interface, pp. 252–260. Calgary, Canada (2002)
28. R.C. Bolles, H.B., Marimont, D.: Epipolar-plane image analysis: An approach to determining structure from motion. International Journal of Computer Vision **1**(1), 7–56 (1987)
29. Ristivojevic, M., Konrad, J.: Joint space-time motion-based video segmentation and occlusion detection using multiphase level sets. In: Visual Communications and Image Processing, vol. 5308, pp. 156–167 (2004)
30. Sato, K., Aggarwal, J.: Temporal spatio-velocity transform and its application to tracking and interaction. Computer Vision and Image Understanding **96**(2), 100–128 (2004)
31. Szeliski, R.: Shape from rotation. In: Proc. IEEE Computer Vision and Pattern Recognition, pp. 625–630. IEEE Computer Society Press (1991)
32. Tang, C.K., Medioni, G.G.: Robust estimation of curvature information from noisy 3d data for shape description. In: International Conference on Computer Vision (ICCV), vol. 1, pp. 426–433 (1999)
33. Taubin, G.: Estimating the tensor of curvature of a surface from a polyhedral approximation. In: International Conference on Computer Vision (ICCV), pp. 902–907 (1995)
34. Zhou, G., Albertz, J., Gwinner, K.: Extracting 3d information using spatio-temporal analysis or aerial image sequence. In: Object Recognition and Scene Classification from Multispectral and Multisensor Pixels, vol. 65, pp. 769–832. Columbus, Ohio, USA (1999)

M. Marquis Bolduc, B.Sc, is currently a M.Sc. student in Computer science at the Université de Sherbrooke (Canada). His research interests concerns video processing and more especially motion interpretation and 3D scene reconstruction.

F. Deschênes received a Ph.D. degree in Computer science (2002) from both the Ecole Nationale Supérieure des Mines de Paris (France) and the Université de Sherbrooke (Canada). He is currently the Dean of research at the Université du Québec en Outaouais (Canada). He is also Professor at Departement of Computer science at the Université de Sherbrooke since 2002. His research interests mainly concern computer vision and more specifically 3D scene understanding, depth cue extraction and video processing.

Conclusion

Le premier article présenté dans ce mémoire, «Combining Apparent Motion and Perspective as Visual Cues for Content-based Camera Motion Indexing», expose une nouvelle méthode pour l'identification des mouvements de la caméra dans une séquence vidéo. Il démontre qu'il est possible d'identifier adéquatement les sept mouvements possibles de la caméra si la séquence contient suffisamment d'informations sur la perspective (via les points de fuites), ce qui n'avait jamais été réalisé. Il s'agit par ailleurs, à notre connaissance, de la première méthode pouvant différencier le changement de la distance focale (zoom) de la translation dans la direction de l'axe optique. La méthode proposée permet d'identifier des mouvements isolés, mais également les combinaisons de ces mouvements.

Le deuxième article, «Occlusion Event Detection using Geometric Features in Spatio-temporal Volumes», décrit une nouvelle méthode pour la détection et la classification des événements d'occlusion. Cette méthode analyse la structure des occlusions dans le volume spatio-temporel et propose d'exploiter cette structure par le bais de la courbure et l'orientation des surfaces spatio-temporelles. Cela permet de reconnaître non seulement les événements d'occultation, mais aussi les contours d'occultation, lesquels sont caractéristiques de tels événements.

Ces deux nouvelles méthodes démontrent le potentiel des indices visuels liés au mouvement dans le but d'interpréter le contenu des séquences vidéo. En effet, ces indices

renseignent sur la manière dont la séquence vidéo a été filmée. De cause à effet, cela peut renseigner sur l'intention et la situation de l'auteur de la séquence, puisque différents mouvements de la caméra sont généralement exploités pour transmettre différentes émotions au spectateur. Il s'agit ainsi d'informations pertinentes pour interpréter le contenu sémantique de la vidéo. Ces indices fournissent par ailleurs de nombreux renseignements sur le comportement et la position des objets dans une séquence vidéo. Par exemple, la méthode proposée peut être utilisée avec une séquence vidéo contenant des occultations, afin de déterminer si un sujet pré-identifié se déplace en avant-plan ou en arrière-plan. La combinaison de ces informations ouvre également de nouvelles perspectives quant à l'identification du sujet de la scène. En effet, le mouvement de la caméra, lorsque présent, suit généralement le sujet afin d'en faire son centre d'intérêt. Il est en effet peu commun de voir le sujet principal d'une scène suivre un déplacement opposé à celui de la caméra, ce qui désorienterait le spectateur. En supposant le sujet localisé, l'identification des contours d'occultation permettrait de déterminer les limites physiques de l'objet dans la scène, et d'en améliorer le suivi.

L'efficacité des deux méthodes proposées fut démontrée à l'aide de séquences vidéos artificielles et réelles, dont certaines tirées du domaine cinématographique, ce qui est au-delà des conditions de laboratoire idéales. Toutefois, il est à noter que les deux méthodes dépendent de la précision de certaines entrées en particulier, la localisation des points de fuites pour la première méthode, et le calcul de la courbure dans un volume pour le second cas. Ces deux sujets, qui font l'objet de recherche[2, 1], permettent l'amélioration de la précision et la fiabilité des deux méthodes proposées.

Bibliographie

[1] Robust estimation of adaptive tensors of curvature by tensor voting. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 27(3):434–449, 2005. W.-S. Tong and C.-K. Tang.

[2] A. Almansa, A. Desolneux, and S. Vamech. Vanishing point detection without any a priori information. *IEEE Transaction of Pattern Analysis and Machine Intelligence*, 25(4):502–507, 2003.

[3] D. Arijon. *Grammar of the Film Language.* Silman-James Press, 1976.

[4] M Marquis Bolduc and F Deschenes. Collision and event detection using geometric features in spatio-temporal volumes. In *CRV '05: Proceedings of the 2nd Canadian conference on Computer and Robot Vision*, pages 236–243, Washington, DC, USA, 2005. IEEE Computer Society.

[5] B. Caprile and V. Torre. Using vanishing points for camera calibration. *International Journal of Computer Vision*, 4(2):127–140, 1990.

[6] J. Coughlan and A. Yuille. The manhattan world assumption: Regularities in scene statistics which enable bayesian inference. In *Neural Information Processing System (NIPS)*, pages 845–851, 2000.

[7] R. Ewerth, M. Schwalb, P. Tessmann, and B. Freisleben. Estimation of arbitrary camera motion in MPEG videos. In *Proceedings of the 17th International Conference on Pattern Recognition (ICPR)*, volume 1, pages 512 – 515, August 23-26 2004.

[8] K. Ikeuchi H. Kawasaki and M. Sakauchi. Spatio-temporal analysis of omni image. In *Computer Vision and Pattern Recognition*, pages II: 577–584, 2000.

[9] P. S. Huggins, H.F. Chen, P. N. Belhumeur, and S. W. Zucker. Finding folds: On the appearance and identification of occlusion. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'01)*, volume 02, page 718, Los Alamitos, CA, USA, 2001. IEEE Computer Society.

[10] K. Kanatani. *Geometric computation for machine vision.* Oxford University Press, Inc., New York, NY, USA, 1993.

[11] D. Koller, J. Weber, and J. Malik. Robust multiple car tracking with occlusion reasoning. Technical report, Berkeley, CA, USA, 1993.

[12] J. Konrad and M. Ristivojevic. Video segmentation and occlusion detection over multiple frames. In *Image and Video Communications and Processing*, pages 377–388, Santa Clara, U.S.A, May 2003.

[13] M. Marquis-Bolduc, F. Deschênes, and W. Pan. Combining apparent motion and perspective as visual cues for content-based camera motion indexing. *Pattern Recognition*, 41(2):445–457, 2008.

[14] O. Monga and S. Benayoun. Using partial derivatives of 3d images to extract typical surface features. *Computer Vision and Image Understanding*, 61(2):171–189, 1995.

[15] S. A. Niyogi. Detecting kinetic occlusion. In *International Conference on Computer Vision*, pages 1044–1049, Cambridge, Massachusetts, USA, 1995.

[16] S.C. Park, H.S. Lee, and S.W. Lee. Qualitative estimation of camera motion parameters from the linear composition of optical flow. *Pattern Recognition*, 37(4):767–779, April 2004.

[17] S.-L. Peng and G. Medioni. Spatio-temporal analysis of an image sequence with occlusion. In *Image Understanding Workshop*, pages 433–442, Boston, Massachusetts, USA, April 1988.

[18] K. van Overveld F. Ernst R. Rodrigues, A. Fernandes. Reconstructing depth from spatiotemporal curves. In *Vision Interface*, pages 252–260, Calgary, Alberta, Canada, 2002.

[19] K. Sato and J.K. Aggarwal. Temporal spatio-velocity transform and its application to tracking and interaction. *Computer Vision and Image Understanding*, 96(2):100–128, November 2004.

[20] M.V. Srinivasan, S. Venkatesh, and R. Hosie. Qualitative estimation of camera motion parameters from video sequences. *Pattern Recognition*, 30(4):593–606, April 1997.

[21] G. Sudhir and J.C.M. Lee. Video annotation by motion interpretation using optical flow streams. *Visual Communication and Image Representation*, 7(4):354–368, December 1996.

[22] J. Y. A. Wang and E.H. Adelson. Representing moving images with layers. *The IEEE Transactions on Image Processing Special Issue: Image Sequence Compression*, 3(5):625–638, September 1994.