Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Alinhamento Espaço-Temporal de Sequências de Vídeo Capturadas a Partir de Múltiplos Pontos de Vista

Flávio Luis Cardeal Pádua

Tese apresentada ao Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Minas Gerais, como requisito parcial para obtenção do título de Doutor em Ciência da Computação.

Orientador: Prof. Rodrigo Lima Carceroni

Belo Horizonte, 20 de maio de 2005

Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Spatio-Temporal Alignment of Video Sequences Captured from Multiple Viewpoints

Flávio Luis Cardeal Pádua

Thesis presented to the Graduate Program in Computer Science of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Doctor in Computer Science.

Advisor: Prof. Rodrigo Lima Carceroni

Belo Horizonte, May 20, 2005

Resumo

Esta tese aborda o problema de se estimar o alinhamento espaço-temporal entre N sequências de vídeo não-sincronizadas referentes à mesma cena dinâmica 3D e capturadas a partir de pontos de vista distintos. Diferentemente dos métodos existentes, os quais funcionam somente para $N = 2$, este trabalho apresenta uma abordagem inovadora que reduz o problema caracterizado por um N qualquer ao problema de se estimar uma única reta em $\mathbb{R}^N$. Esta reta captura todas as relações temporais entre os videos, podendo ser calculada sem qualquer conhecimento *a priori* sobre as mesmas. Considerando que o alinhamento espacial é capturado por parâmetros de tensores bilineares (matrizes fundamentais), um algoritmo iterativo é usado para refinar simultaneamente os parâmetros temporais e espaciais que definem o alinhamento entre as sequências, uma vez que o refinamento exclusivo dos parâmetros temporais é subótimo. Resultados experimentais obtidos com sequências de vídeo reais demonstram que a metodologia proposta é capaz de recuperar eficazmente o alinhamento entre as sequências mesmo diante da existência de grandes desalinhamentos, diante da presença de ambiguidades (por exemplo, cenas com movimentos periódicos) e quando um alinhamento manual preciso é inviável. Finalmente, experimentos com sequências sintéticas demonstram a escalabilidade e acurácia de nossa abordagem, fornecendo medidas quantitativas para a qualidade dos alinhamentos estimados.

Abstract

This thesis addresses the problem of estimating the spatio-temporal alignment between N unsynchronized video sequences of the same dynamic 3D scene, captured from distinct viewpoints. Unlike existing methods, which work for $N = 2$ and rely on a computationally-intensive search in the space of temporal alignments, we present a novel approach that reduces the problem for general N to the robust estimation of a single line in $\mathbb{R}^N$. This line captures all temporal relations between the sequences and can be computed without any prior knowledge of these relations. Considering that the parameters of fundamental matrices capture the spatial alignment, we use an iterative algorithm to refine simultaneously the parameters representing the temporal and spatial relations between the sequences, since that the exclusive refinement of the temporal parameters is suboptimal. Experimental results with real-world sequences show that our method can accurately align videos even when they have large misalignments (e.g., hundreds of frames), when the problem is seemingly ambiguous (e.g., scenes with roughly periodic motion), and when accurate manual alignment is difficult (e.g., due to slow-moving objects). Finally, experiments with synthetic sequences demonstrate the scalability and accuracy of our approach, providing quantitative measurements for the quality of the spatio-temporal alignments estimated.

Resumo Estendido

O texto a seguir consiste em um resumo estendido sobre o trabalho desenvolvido nesta tese. Primeiramente, este texto introduz o problema abordado, a principal motivação para se resolvê-lo e alguns dos principais trabalhos relacionados. Em seguida, é feita uma breve descrição da metodologia desenvolvida e dos experimentos realizados que comprovam sua aplicabilidade, escalabilidade e exatidão. Finalmente, são apresentadas conclusões e propostas de trabalhos futuros.

Introdução

Esta tese aborda o problema de se estimar o *Alinhamento Espaço-Temporal entre Múltiplas Sequências de Vídeo* referentes a uma mesma cena 3D, as quais são capturadas a partir de pontos de vista distintos. A dinâmica da cena bem como características estáticas presentes na mesma são utilizadas como poderosas pistas para se estimar a *sincronização temporal* (alinhamento temporal) e o *alinhamento espacial* entre as sequências. Tipicamente, o *desalinhamento temporal* entre sequências de vídeo origina-se por duas razões principais. A primeira relaciona-se com o fato de que as sequências de entrada podem possuir diferentes taxas de quadros, enquanto a segunda relaciona-se com a existência de um deslocamento temporal entre as sequências frequentemente criado quando as câmeras não são ativadas simultaneamente. Por outro lado, o *desalinhamento espacial* resulta das diferentes posições, orientações e parâmetros internos de calibração das câmeras.

Em muitas aplicações atuais que se beneficiam da disponibilidade de registros de vídeo simultâneos de um mesmo evento físico, como por exemplo, tele-imersão (Vedula et al., 2002), segurança baseada em vídeo (Zelnik-Manor and Irani, 2001), criação de mosaicos a partir de múltiplos vídeos (Caspi and Irani, 2001) e análise de lances duvidosos em eventos esportivos (Reid and Zisserman, 1996), observa-se a necessidade da estimação numa fase anterior do *alinhamento espaço-temporal* das múltiplas sequências de vídeo referentes ao evento físico monitorado.

Neste contexto, nota-se uma demanda crescente por métodos automáticos eficazes para a estimação do alinhamento espaço-temporal entre múltiplas sequências, especialmente sequências previamente gravadas onde o uso de hardwares de sincronização é inviável. Sendo assim, esta tese propõe uma nova abordagem cujo objetivo principal consiste em avançar no desenvolvimento de novas metodologias para se estimar com grande exatidão o alinhamento espaço-temporal entre não somente duas sequências, como a maioria dos métodos existentes, mas entre um conjunto genérico de N sequências.

Especificamente, este trabalho considera conjuntos de sequências que possuam uma certa sobreposição entre seus campos de visão, isto é, dadas N sequências, é possível identificar tuplas correspondentes de pixels (x_1, y_1, t_1) $\in S_1$,...,$(x_N, y_N, t_N) \in S_N$, onde todas estas tuplas são formadas pelas projeções de um único ponto na cena. Além disso, a abordagem apresentada neste trabalho considera que todas as regiões de sobreposição contenham algum movimento não-rígido.

Nós acreditamos que qualquer solução suficientemente genérica para o problema de alinhamento espaço-temporal deva tratar os seguintes casos:

- **Taxas de quadros desconhecidas:** as taxas de quadros das sequências são desconhecidas e podem apresentar qualquer valor.
- **Deslocamento temporal arbitrário:** o deslocamento temporal entre as sequências é desconhecido e pode ser arbitrariamente grande.
- **Movimento desconhecido:** o movimento 3D dos objetos na cena é desconhecido e suas características são arbitrárias.
- **Falhas no rastreamento:** pontos de interesse na cena não podem ser rastreados de forma confiável ao longo de toda a sequência.
- **Geometria epipolar desconhecida:** a geometria epipolar das câmeras de vídeo é desconhecida.
- **Escalabilidade:** a eficiência computacional da metodologia utilizada deve cair proporcionalmente ao aumento no número de sequências.
- **Ausência de pontos estáticos:** nenhum ponto visível na cena permanece estático para dois ou mais quadros.

Neste sentido, esta tese apresenta uma abordagem inovadora que trata todos os casos acima mencionados, com exceção do último caso. Em particular, nós assumimos que para cada par de sequências de vídeo é possível identificar pontos estáticos suficientes na cena que permitam a obtenção de uma estimativa inicial da geometria epipolar das câmeras. Além disso, com o objetivo de se assegurar que os parâmetros desta estimativa inicial permaneçam constantes durante a aplicação de nossa abordagem, nós consideramos um cenário no qual as câmeras são estáticas e apresentam parâmetros intrínsicos e extrínsicos constantes e desconhecidos.

A idéia básica de nossa abordagem está fundamentada na definição de uma reta N-dimensional que captura globalmente as relações temporais entre todas as N sequências de vídeo. Uma propriedade fundamental desta reta é que ainda embora seu conhecimento implique no conhecimento do alinhamento temporal entre as sequências, a estimativa de pontos sobre a mesma pode ser realizada sem o conhecimento prévio de tal alinhamento. Utilizando-se esta propriedade como ponto de partida, a abordagem proposta neste trabalho reduz o problema de se estimar o alinhamento temporal entre N sequências para o problema de se estimar de forma robusta uma única reta N-dimensional a partir de um conjunto de pontos gerados em $\Re^N$.

Grande parte das soluções existentes na literatura para o problema de se adquirir o alinhamento temporal realizam uma pesquisa explícita em todo o espaço de soluções de alinhamentos possíveis (Caspi et al., 2002; Rao et al., 2003; Wolf and Zomet, 2002a,b; Lee et al., 2000; Stein, 1998). Infelizmente, a natureza combinatória desta pesquisa requer o estabelecimento de várias hipóteses adicionais para torná-la gerenciável, como por exemplo, deve-se conhecer *a priori* as taxas de quadros das sequências, deve-se assumir sempre $N = 2$, que o desalinhamento temporal é um inteiro e ainda que tal desalinhamento resida dentro de uma pequena faixa limitada definida pelo usuário (tipicamente menor do que cinquenta quadros). Consequentemente, ainda que a maior parte destas soluções possam ser utilizadas diante da inexistência de pontos estáticos na cena, suas eficiências computacionais podem limitar bastante suas aplicabilidades. Diferentemente de tais técnicas, nossa abordagem alinha N sequências num único passo, pode tratar desalinhamentos temporais arbitrariamente grandes e não requer qualquer informação *a priori* sobre as relações temporais entre as mesmas.

A metodologia proposta nesta tese está mais diretamente relacionada com a abordagem proposta em Caspi et al. (2002). Neste trabalho, a técnica proposta pelos autores recupera a geometria epipolar e o desalinhamento temporal entre as sequências a partir da trajetória no plano de imagem de um único ponto na cena que é visível em ambas as sequências. Posteriormente, a geometria epipolar e o desalinhamento temporal estimados são refinados usando-se mais pontos. Para fazerem isso, os autores assumem que as taxas de quadros são conhecidas e formulam um problema de otimização não-linear para estimar os parâmetros refinados que capturam a geometria epipolar e o desalinhamento temporal. Infelizmente, a natureza altamente não-linear deste processo de otimização necessita da aquisição de boas estimativas iniciais para a geometria epipolar e o desalinhamento temporal. É importante ainda mencionar que a abordagem proposta em Caspi et al. (2002) assume que um único ponto na cena possa ser rastreado de forma confiável ao longo de toda a sequência, o que pode ser difícil de se realizar no caso de vídeos de cenas complexas, onde o rastreamento pode falhar devido a diversos problemas, como por exemplo, problemas de oclusão. Nossa solução, por outro lado, requer a habilidade de se rastrear pontos na cena somente ao longo de dois quadros consecutivos da mesma sequência. Além disso, ela não requer a habilidade de se estabelecer correspondência de pontos entre as sequências.

A seguir será apresentada uma breve descrição da metodologia desenvolvida nesta tese, a qual pode ser dividida em duas etapas principais. Em sua primeira etapa, chamada de *Alinhamento Temporal*, realiza-se uma estimativa inicial dos parâmetros que recuperam o alinhamento temporal entre as sequências, utilizando-se para isso as trajetórias de pontos de interesse que se movem na cena bem como as estimativas iniciais das matrizes fundamentais que recuperam o alinhamento espacial entre as sequências. Na segunda e última etapa de nossa metodologia, chamada de *Alinhamento*

Espaço-Temporal, um processo de otimização linear é formulado para se refinar os parâmetros temporais e espaciais estimados inicialmente, os quais podem possuir erros significativos relacionados com as limitações impostas pelas técnicas utilizadas para se recuperar as estimativas iniciais das geometrias epipolares de pares de câmeras bem como pelas técnicas utilizadas para se adquirir as trajetórias dos objetos móveis.

Alinhamento Temporal

Considere uma cena dinâmica monitorada simultaneamente por N câmeras localizadas em diferentes pontos de vista, onde tais câmeras obedecem ao modelo de projeção em perspectiva. Assuma que cada câmera capture quadros a uma taxa constante e desconhecida. Além disso, considere que as câmeras não estejam sincronizadas, isto é, elas começam a capturar quadros em diferentes instantes de tempo e possivelmente com taxas de quadros distintas. Com o objetivo de se alinhar temporalmente as sequências de vídeo resultantes, deve-se determinar as correspondências entre os números dos quadros de uma sequência de "referência" com os números dos quadros em todas as outras sequências. Esta correspondência pode ser expressa como um conjunto de equações lineares do tipo:

$$f_i = \alpha_i f_r + \beta_i, \tag{1}$$

onde f_i e f_r são os números dos quadros da i-ésima sequência e da sequência de referência, respectivamente, e α_i, β_i são constantes desconhecidas que representam a dilatação e o deslocamento temporal, respectivamente, entre as sequências. Em geral, estas constantes não são números inteiros.

As relações temporais entre pares de sequências capturadas pela Equação

(3.1) induzem uma relação global entre os números dos quadros das sequências de entrada. Especificamente, nós representamos esta relação global pela reta N-dimensional $\mathcal{L}$ a seguir:

$$\mathcal{L} = \left\{ [\alpha_1 \ldots \alpha_N]^T t + [\beta_1 \ldots \beta_N]^T \quad | \quad t \in \Re \right\}. \tag{2}$$

A propriedade chave desta reta N-dimensional, pela qual a estimativa de pontos sobre a mesma pode ser realizada sem o conhecimento prévio do alinhamento temporal entre as sequências, permite a elaboração de um algoritmo simples para sua reconstrução a partir das trajetórias de características de interesse que se movem na cena e são visíveis por duas ou mais sequências.

Especificamente, seja $\mathbf{q}_1(f_1)$ a projeção instantânea na câmera 1 no quadro f_1 de um ponto móvel na cena, expressa em coordenadas 2D homogêneas, como ilustrado na Figura 3.1. Além disso, seja $\mathbf{q}_i(\cdot)$ a trajetória formada pela projeção deste ponto móvel na câmera i e suponha que a matrix fundamental $\mathcal{F}_{1i}$ entre as câmeras 1 e i sejam conhecidas para todo i, onde $i = 2...N$. Neste caso, a câmera 1 está sendo considerada como sendo a câmera de referência. Se o ponto de interesse na cena é visível por todas as câmeras quando o quadro f_1 é capturado pela câmera 1, tem-se a seguinte restrição:

> O conjunto:
>
> $$\mathcal{V}_{\mathbf{q}_1(f_1)} = \left\{ [f_1 \ldots f_N]^T \;\; | \;\; \mathbf{q}_1^\top(f_1)\mathcal{F}_{1i}\mathbf{q}_i(f_i) = 0, \; i = 2 \ldots N \right\}$$
>
> contém pelo menos um ponto sobre a reta N-dimensional $\mathcal{L}$.

Intuitivamente, a restrição acima pode ser imaginada como um procedimento para se gerar um conjunto $\mathcal{V}_{\mathbf{q}_1(f_1)}$ de alinhamentos temporais "candidatos", o qual deve conter pelo menos um ponto sobre a reta N-dimensional procurada. Além disso, esta restrição nos diz que tal conjunto pode ser criado de acordo com os seguintes passos principais: (1) intersectar a reta epipolar de $\mathbf{q}_1(f_1)$ com a trajetória $\mathbf{q}_i(\cdot)$ na câmera i, (2) armazenar o(s) número(s) dos quadro(s) correspondente(s) ao(s) ponto(s) de interseção na câmera i e (3) gerar vetores de alinhamento temporal a partir dos números dos quadros armazenados.

Para que se possa aplicar a restrição acima apresentada, deve-se conhecer *a priori* as matrizes fundamentais $\mathcal{F}_{ij}$, as quais descrevem a geometria epipolar de cada par de câmeras (i, j). Na prática, nossa abordagem obtém uma estimativa inicial de $\mathcal{F}_{ij}$ por meio da identificação de pontos estáticos no plano de fundo da cena, os quais sejam visíveis por duas ou mais câmeras. Uma vez que a reta N-dimensional $\mathcal{L}$ tenha sido reconstruída, isto é, uma vez que a estimação do alinhamento temporal entre as sequências tenha sido realizada, nossa abordagem realiza um processo de otimização linear dos parâmetros de $\mathcal{L}$ juntamente com os parâmetros das matrizes fundamentais que descrevem a geometria da cena. A seguir, será descrito o algoritmo proposto nesta tese para se reconstruir $\mathcal{L}$.

Recontrução da reta N-dimensional $\mathcal{L}$

Pode-se notar que a restrição descrita na seção anterior nos leva a um algoritmo baseado em um processo de votação para se reconstruir a reta N-dimensional que recuperará o alinhamento temporal entre N sequências. Especificamente, este algoritmo opera em duas etapas principais. Na primeira etapa, escolhe-se uma das sequências de vídeo como sendo a sequência de "referência" e utiliza-se as posições instantâneas $\mathbf{q}_r(f_r)$ de cada trajetória $\mathbf{q}_r(\cdot)$ desta sequência de referência juntamente com as trajetórias completas $\mathbf{q}_i(\cdot)$ das outras sequências com o objetivo de se estimar $\mathcal{V}_{\mathbf{q}_r(f_r)}$ para cada $\mathbf{q}_r(f_r)$. Na segunda etapa, estima-se a reta N-dimensional $\mathcal{L}$ a partir da nuvem de pontos formada pela união dos conjuntos $\mathcal{V}_{\mathbf{q}_r(f_r)}$. Sendo assim, para especificarmos completamente este algoritmo, três questões principais devem ser respondidas:

1. Como são calculadas as trajetórias $\mathbf{q}_i(\cdot)$ dos pontos de interesse na cena, para $i = 1, ..., N$?

2. Como é estimado o conjunto $\mathcal{V}_{\mathbf{q}_r(f_r)}$ para cada $\mathbf{q}_r(f_r)$?

3. Como são calculados os parâmetros de $\mathcal{L}$?

Para se calcular as trajetórias $\mathbf{q}_i(\cdot)$, nossa metodologia utiliza um rastreador de características, o qual é tratado por nosso algoritmo como uma "caixa preta" responsável por retornar uma lista de segmentos de reta de características correspondentes para todo par de quadros consecutivos. É importante ressaltar que nosso algoritmo independe do rastreador utilizado. Assim, a escolha de uma determinada metodologia de rastreamento depende somente da complexidade da cena e das propriedades das características de interesse.

No que se refere ao cálculo do conjunto $\mathcal{V}_{\mathbf{q}_r(f_r)}$ para um dado $\mathbf{q}_r(f_r)$, nosso algoritmo utiliza as estimativas iniciais das matrizes fundamentais $\mathcal{F}_{ij}$, entre cada par (i, j) de câmeras, assim como também os segmentos de reta fornecidos pelo rastreador. Quando um segmento de reta específico intersecta a reta epipolar de $\mathbf{q}_r(f_r)$, define-se um número de quadro possivelmente não inteiro f_i, o qual representa a i-ésima coordenada de um elemento potencial de $\mathcal{V}_{\mathbf{q}_r(f_r)}$. Para se gerar $\mathcal{V}_{\mathbf{q}_r(f_r)}$, agrupa-se todas as coordenadas f_i calculadas para todas as sequências e concatena-se as mesmas de tal forma que elas construam vetores N-dimensionais válidos, os quais representam alinhamentos temporais candidatos em um espaço de votos. Estes passos são ilustrados na Figura 3.4(a)-(d) para o cojunto de duas sequências de vídeo reais exibidas na Figura 5.1.

Observe que se a reta epipolar de $\mathbf{q}_r(f_r)$ intersecta dois ou mais segmentos de reta em cada uma das $N-1$ câmeras restantes, tem-se um total de 2^{N-1} possíveis maneiras de se concatenar em um vetor N-dimensional as coordenadas f_i estimadas. Para se evitar a inclusão de um número exponencial de vetores em $\mathcal{V}_{\mathbf{q}_r(f_r)}$, nosso algoritmo somente inclui aqueles que sejam consistentes com a geometria epipolar das câmeras. Em particular, concatena-se as coordenadas f_i e f_j para as câmeras i e j, respectivamente, se os pontos de interseção que as definiram estão próximos de suas retas epipolares correspondentes. Observe que nosso procedimento de concatenação é conservador, isto é, ele garante que o conjunto de vetores gerado desta maneira será um superconjunto de $\mathcal{V}_{\mathbf{q}_r(f_r)}$.

Em geral, o conjunto de todos os alinhamentos temporais candidatos contém um grande número de dados espúrios (do inglês, *outliers*), como ilustrado na Figura 3.4(d). Neste contexto, para se reconstruir a reta N-dimensional $\mathcal{L}$, nossa metodologia se baseia no uso de um algoritmo bastante robusto a presença de tais dados, conhecido como RANSAC (Fischler and Bolles, 1981), o qual é descrito detalhadamente no Apêndice A. Basicamente, o algoritmo RANSAC escolhe aleatoriamente um par de alinhamentos temporais candidatos para se definir a reta $\mathcal{L}$ e, em seguida, calcula o número total de candidatos que se encontram a uma distância máxima ϵ desta reta. Estes dois passos são repetidos durante um número determinado de iterações. Portanto, os dois parâmetros críticos deste algoritmo são o número z de iterações e a distância ϵ. Para se determinar z, utiliza-se a seguinte equação:

$$z = \left\lceil \frac{\log(1-p)}{\log(1-r^2)} \right\rceil , \tag{3}$$

onde p é um parâmetro especificado *a priori* cujo valor varia de 0 a 1 e r é a probabilidade de um candidato aleatoriamente selecionado ser um ponto sobre $\mathcal{L}$ (do inglês, *inlier*). A Equação (3.3) expressa o fato de que z deva ser suficientemente grande para assegurar que, com probabilidade p, pelo menos um dos pares de candidatos selecionados aleatoriamente esteja sobre $\mathcal{L}$. Em nossos experimentos, foram utilizados $p = 0.99$ e $r = 0.05$, os quais são valores conservadores que levaram aos resultados mais exatos de alinhamento espaço-temporal. A distância máxima ϵ é calculada pela distância média entre características estáticas identificadas nas sequências de entrada e suas respectivas retas epipolares.

Finalmente, após a estimativa realizada pelo RANSAC do potencial conjunto de candidatos sobre os quais a reta $\mathcal{L}$ deva passar, nossa metodologia utiliza o método dos mínimos quadrados sobre este conjunto para se estimar os parâmetros de $\mathcal{L}$. A seguir, será apresentada a segunda etapa de nossa metodologia, a qual consiste na realização de um processo de otimização dos parâmetros temporais e espaciais estimados na primeira etapa.

Alinhamento Espaço-Temporal

Embora as imagens de uma cena dinâmica possam conter pontos estáticos em seus planos de fundo, observa-se que tais pontos frequentemente não representam a maior parte das características detectáveis na cena. Qualquer metodologia que busque estimar a geometria epipolar entre um par de câmeras exclusivamente a partir deste conjunto de características estáticas estará provavelmente ignorando um conjunto significativamente grande de informações relevantes disponíveis nas imagens. Na prática, esta conduta poderá causar erros nas matrizes fundamentais calculadas e, em última análise, na reta N-dimensional $\mathcal{L}$ calculada por nosso método. A seguir, será brevemente mostrado como nossa metodologia refina as matrizes fundamentais $\mathcal{F}_{ij}$ e os parâmetros de $\mathcal{L}$ por meio da consideração de todas as características detectadas nas sequências. Sem perda de generalidade, a câmera 1 é assumida como sendo a câmera de referência.

Seja $\mathbf{q}_1(f_1)$ a projeção de um ponto da cena no plano de imagem da câmera 1 durante a aquisição do quadro f_1. Além disso, suponha que a projeção instantânea de tal ponto da cena em uma câmera i no quadro f_i possa ser parametrizada como a seguir:

$$\mathbf{q}_i(f_i) = (1 - f_i)\mathbf{q}_i(f_a) + f_i\mathbf{q}_i(f_b), \tag{4}$$

onde $\mathbf{q}_i(f_a)$ e $\mathbf{q}_i(f_b)$ são os extremos de um segmento linear, o qual contém a posição $\mathbf{q}_i(f_i)$. Observe que esta equação permite estabelecer uma restrição envolvendo os parâmetros de $\mathcal{L}$ e os parâmetros da matriz fundamental $\mathcal{F}_{1i}$. Especificamente, pode-se utilizar os parâmetros de $\mathcal{L}$ para se calcular a posição instantânea $\mathbf{q}_i(f_i)$ correspondente à posição instantânea $\mathbf{q}_1(f_1)$ na câmera 1, como se segue:

$$\mathbf{q}_i(f_i) = [1 - (\alpha_i f_1 + \beta_i)\mathbf{q}_i(f_a)] + (\alpha_i f_1 + \beta_i)\mathbf{q}_i(f_b). \tag{5}$$

Além disso, sendo $\mathbf{q}_i(f_i)$ e $\mathbf{q}_1(f_1)$ pontos correspondentes, tem-se que:

$$\mathbf{q}_1^\top(f_1)\mathcal{F}_{1i}\mathbf{q}_i(f_i) = 0. \tag{6}$$

Combinando-se as Equações (5) e (6), obtém-se uma equação homogênea que leva em conta os parâmetros estimados de $\mathcal{L}$, isto é, os parâmetros α_i, β_i, bem como os parâmetros estimados da matriz fundamental $\mathcal{F}_{1i}$:

$$(1 - \alpha_i f_1 - \beta_i)\mathbf{q}_1^\top(f_1)\mathcal{F}_{1i}\mathbf{q}_i(f_a) + (\alpha_i f_1 + \beta_i)\mathbf{q}_1^\top(f_1)\mathcal{F}_{1i}\mathbf{q}_i(f_b) = 0. \tag{7}$$

Na prática, erros nos parâmetros de $\mathcal{L}$ e $\mathcal{F}_{1i}$ fazem com que a Equação (7) não seja satisfeita exatamente, originando-se um resíduo que representa a distância algébrica entre $\mathbf{q}_i(f_i)$ e sua reta epipolar correspondente. Para se refinar a estimativa inicial dos parâmetros temporais e espaciais, nossa técnica realiza a expansão da Equação (7) ao introduzir as incógnitas $\Delta\mathcal{F}_{1i}$, $\Delta\alpha_i$ e $\Delta\beta_i$ na mesma, de forma que:

$$\begin{aligned}\mathcal{F}_{1i} &\longleftarrow \mathcal{F}_{1i} + \Delta\mathcal{F}_{1i}\\ \alpha_i &\longleftarrow \alpha_i + \Delta\alpha_i\\ \beta_i &\longleftarrow \beta_i + \Delta\beta_i.\end{aligned}$$

Dada a equação expandida, determina-se as incógnitas $\Delta\mathcal{F}_{1i}$, $\Delta\alpha_i$ e $\Delta\beta_i$ que minimizam seu resíduo. Note que para cada ponto na cena e cada quadro no qual ele é visível, nossa técnica obtém uma equação em função dos parâmetros desconhecidos que especificam completamente $\Delta\mathcal{F}_{1i}$, $\Delta\alpha_i$ e $\Delta\beta_i$. Esta equação é não linear pelo fato de conter os termos de segunda ordem $(\Delta\alpha_i\Delta\mathcal{F}_{1i})$ e $(\Delta\beta_i\Delta\mathcal{F}_{1i})$. Em nossa implementação, nós simplificamos os cálculos a serem realizados ao ignorarmos estes termos não lineares e resolvermos o sistema resultante sobredeterminado de equações lineares.

Em geral, as trajetórias de um objeto móvel na cena são altamente não lineares. Para superarmos este problema, a técnica de refinamento acima descrita é aplicada a um conjunto de segmentos de reta que aproximam a trajetória original realizada pelo ponto na cena (Horst and Beichl, 1997), como ilustrado na Figura 4.1. A seguir são apresentados os resultados experimentais que comprovam a eficácia da metodologia proposta neste trabalho.

Resultados experimentais

Os experimentos realizados neste trabalho são divididos em dois grupos principais. No primeiro grupo, realizamos experimentos com sequências de vídeo reais para avaliarmos e demonstrarmos a aplicabilidade de nossa abordagem, sobretudo em cenários desafiadores para os demais métodos encontrados na literatura, como por exemplo, cenários nos quais o sistema de rastreamento não é confiável ao longo de toda a sequência e estimativas iniciais exatas para as matrizes fundamentais não são possíveis. Por outro lado, para avaliarmos cuidadosamente a escalabilidade e exatidão de nossa metodologia diante da variação de alguns parâmetros críticos para a qualidade dos alinhamentos espaço-temporais calculados, um segundo grupo de experimentos com sequências sintéticas de uma cena artificial foi realizado. A seguir, os resultados experimentais obtidos são brevemente descritos.

Sequências reais

As sequências de vídeo reais utilizadas em nossos experimentos contém dois e três pontos de vista distintos do evento físico monitorado, sendo representantes de cenários bastante diversos onde, por exemplo, as sequências possuem comprimentos que variam de 55 a 605 quadros, os desalinhamentos temporais variam de 21 a 285 quadros, as imagens podem apresentar desde uma alta qualidade (sequências capturadas em laboratório) a uma baixa qualidade (video clipes de transmissões de TV), as características de interesse podem se mover desde vários pixels a menos do que um pixel por quadro e as razões entre as taxas de quadros de pares de sequência varia entre 1 e 2. Em todos os casos, as dimensões das imagens são 320 × 240 pixels. A exatidão dos alinhamentos temporais calculados é avaliada com base no cálculo do erro de alinhamento temporal médio. Este erro é dado pela média das diferenças absolutas entre as coordenadas temporais calculadas utilizando-se a reta $\mathcal{L}$ estimada por nosso método e as coordenadas temporais calculadas usando-se a reta $\mathcal{L}$ de referência, a qual é determinada a partir de um processo de alinhamento manual.

Sequência 1

Como um primeiro experimento, nossa técnica de alinhamento espaço-temporal foi aplicada a duas sequências de vídeo (veja Figura 5.1) também utilizadas em Caspi and Irani (2000). Basicamente, o objetivo deste experimento inicial era comparar a eficácia de nossa técnica com a eficácia da metodologia desenvolvida por estes autores. As duas sequências adquirem quadros a uma taxa de 25qps ($\alpha = 1$) e possuem um desalinhamento temporal de referência dado por: $\beta = 55 \pm 0.5$ quadros.

A Figura 3.4(d) exibe a reta $\mathcal{L}$ reconstruída utilizando-se a primeira etapa de nossa metodologia. Esta estimativa inicial de $\mathcal{L}$ leva a um erro de alinhamento temporal médio de 0.66 quadros. Entretanto, após a aplicação de nossa técnica de refinamento, a nova reta estimada levou a um erro ainda menor de 0.35 quadros. Note que os parâmetros calculados por nosso método são tão exatos quanto aqueles calculados em Caspi and Irani (2000), embora nossa metodologia se baseie na solução de um problema muito menos restritivo (por exemplo, α é desconhecido e a cena não precisa ser planar).

Sequência 2

O segundo experimento foi realizado com duas sequências de mesma taxa de quadros (30qps) adquiridas em laboratório. Neste caso, os objetivos principais eram mostrar que nossa metodologia independe do fato da cena ser ou não planar e do tamanho do desalinhamento temporal entre as sequências. Especificamente, a cena monitorada é constituída por objetos (robôs) que se movem em dois planos distintos, como ilustrado na Figura 5.3. Os parâmetros temporais são $\alpha = 1$ e $\beta = -284.5 \pm 2$ quadros. Observe que o desalinhamento temporal entre as sequências é significativamente grande.

A reta $\mathcal{L}$ inicialmente reconstruída é mostrada na Figura 5.4, a qual leva a um erro de alinhamento temporal médio de 5.84 quadros. Entretanto, após a aplicação da técnica de refinamento, a nova reta estimada diminuiu este erro para 4.43 quadros. Dado que os robôs se movem muito lentamente, este erro de alinhamento temporal aparentemente significativo não é perceptível visualmente. A Figura 5.7 confirma a boa qualidade do alinhamento temporal recuperado.

Sequência 3

Neste experimento, duas pessoas são monitoradas por um par de câmeras com mesma taxa de quadros (30qps), enquanto realizam um malabarismo com cinco bolas coloridas em suas mãos (veja Figura 5.5). Estas sequências representam um cenário bastante difícil para todos os métodos existentes, uma vez que (1) as trajetórias das diferentes bolas se sobrepõe no mundo 3D, (2) as trajetórias individuais são aproximadamente cíclicas e (3) as bolas se movem muito rapidamente (mais de 9 pixels por quadro). Os parâmetros temporais são $\alpha = 1$ e $\beta = -41 \pm 0.5$ quadros. Para tornar o problema de se estimar o alinhamento temporal ainda mais desafiador, uma das sequências de entrada foi inicialmente modificada por meio da inserção de novos quadros e, posteriormente, por meio da exlcusão de quadros. Estas modificações foram realizadas para se simular sequências que apresentam mais do que uma taxa de quadros e sequências que possuem vídeo clipes intermediários, como por exemplo, comerciais de TV.

As Figuras 5.6(a)-(c) mostram as retas reconstruídas antes do refinamento, as quais capturam os alinhamentos temporais entre as sequências. Para as sequências originais, obtivemos erros de alinhamento temporal de 0.75 e 0.26 quadros antes e após o refinamento, respectivamente.

Sequência 4

Neste último experimento com sequências reais, nossa técnica foi aplicada a três sequências de vídeo referentes a uma partida de futebol (veja Figura 5.9). As sequências apresentam baixa qualidade e foram adquiridas a partir de câmeras que se moviam. Neste contexto, para que pudéssemos utilizar nossa metodologia, foi preciso compensarmos os movimentos das câmeras numa fase anterior (Brown and Lowe, 2003).

Uma vez que agora três sequências de vídeo são consideradas, a reta $\mathcal{L}$ calculada por nossa metodologia é uma reta 3D, como ilustrada nas Figuras 5.10(b) and 5.10(c). Uma percepção visual da boa qualidade do alinhamento temporal estimado por nossa técnica é fornecida pela Figura 5.11.

Sequências sintéticas

Na segunda etapa de nossos experimentos, um software foi desenvolvido para se simular as cinemáticas de partículas 3D com movimentos independentes, bem como a visualização das mesmas a partir de múltiplos pontos de vista, gerando-se assim sequências de vídeo sintéticas. Por meio da utilização de tais dados sintéticos, foi possível controlar e avaliar os impactos de valores específicos atribuídos a parâmetros considerados críticos para a exatidão e escalabilidade de nossa metodologia.

Em particular, com estes experimentos foram analisados os efeitos da variação de três parâmetros principais: (1) número de objetos rastreados, (2) erro na geometria epipolar de cada par de câmeras e, finalmente, (3) o nível de ruído do sistema de rastreamento.

A cena artificial gerada simula a cena ilustrada na Figura 5.12(a), a qual contém duas câmeras calibradas, cujos parâmetros intrínsecos e extrínsicos são aqueles das câmeras utilizadas nos experimentos com sequências reais obtidas em laboratório. As trajetórias das partículas aleatórias eram geradas no interior de uma esfera 3D, cujo raio foi definido empiricamente de forma a maximizar suas projeções nos planos de imagem de ambas as câmeras (veja Figura 5.12(b)). Cada partícula era iniciada aleatoriamente em uma posição uniformemente distribuída no interior desta esfera. O modelo utilizado para a cinemática de uma partícula específica é ilustrado na Figura 5.13.

Considerando-se conjuntos de k partículas ($k \in \{1, 2, 4, 8, 16, 32\}$), dois cenários principais foram concebidos para a realização dos experimentos com sequências sintéticas. Primeiramente, considerou-se um cenário no qual a geometria epipolar do par de câmeras continha um erro fixo de 2 pixels e variou-se o desvio padrão R (em pixels) do ruído gaussiano de média zero adicionado ao rastreador ($R \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$). Por outro lado, no segundo cenário, fixou-se o desvio padrão do ruído do rastreador em 2 pixels e variou-se o erro ε_f da matriz fundamental ($\varepsilon_f \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$). Fixando-se o erro da matriz fundamental e o desvio padrão do ruído do rastreador em 2 pixels, simulou-se situações mais realísticas, uma vez que tal valor foi o pior caso para as matrizes fundamentais calculadas para as sequências reais e representa na média o desvio padrão do ruído observado em alguns dos rastreadores usados na prática. Para cada tupla (ε_f, R, k)

em cada um dos dois cenários acima descritos, foram simulados 100 eventos dinâmicos distintos na cena artificial. Em seguida, dados os espaços de votos dos eventos dinâmicos simulados, foram calculados os percentuais das retas que capturam o alinhamento temporal entre as sequências que levam a erros médios de alinhamento temporal menores ou iguais a 1, 2 e 5 quadros.

Analisando-se os resultados obtidos e supondo-se que o limite mínimo desejável para o percentual de retas que levam a um determinado erro médio de alinhamento temporal seja 95%, nota-se que para valores de erro na geometria epipolar e no sistema de rastreamento próximos àqueles verificados na prática, nossa metodologia não somente alcançou como também superou o limite desejável de 95% para diversos valores de número de objetos rastreados, como ilustrado na Figura 5.16(a). Observe nesta figura que o uso da técnica de refinamento mostrou-se ser de grande importância, sendo que a melhoria trazida aos resultados pela mesma caiu bruscamente quando se aumentou o erro no sistema de rastreamento e na geometria epipolar das câmeras (por exemplo, veja Figura 5.20(d), onde $R = 10$).

Considerando-se os valores de erro na geometria epipolar e no rastreador que foram efetivamente observados na prática ($\varepsilon_f = R = 2$ pixels), observa-se a partir das Figuras 5.30(a) e 5.33(a) que para aplicações onde se necessita de alinhamentos temporais bastante exatos ($\varepsilon_f \leq 1$ quadro), nossa metodologia alcançou um percentual de sucessos de 60%. Por outro lado, considerando-se aplicações mais flexíveis quanto ao valor do erro de alinhamento temporal (por exemplo, $\varepsilon_f \leq 5$ quadros), nota-se que até mesmo para rastreadores bastante corrompidos por ruído ($R = 10$) e matrizes fundamentais com erros severos ($\varepsilon_f = 4$), nossa metodologia superou o limite desejável de 95% para o percentual de sucessos, como ilustrado nas Figuras 5.30(c) e 5.33(c).

Finalmente, nota-se a partir de nossos resultados que para conjuntos muito pequenos de objetos rastreados (por exemplo, $k = 1$) ou conjuntos muito grandes (por exemplo, $k = 32$), os percentuais de sucesso de nossa metodologia diminuem significativamente, como exemplificado na Figura 5.20(d)-(f). As razões para tal fato estão relacionadas com os volumes de pontos gerados nos correspondentes espaços de votos. Por exemplo, observe a Figura 5.14(a)-(f). Quando se tem um pequeno número de objetos rastreados (veja Figura 5.14(a)), obtém-se consequentemente um espaço de votos com poucos pontos e, em particular, poucos *inliers*, o que dificulta o trabalho do RANSAC para encontrar a reta $\mathcal{L}$ desejada. Por outro lado, quando um número muito grande de objetos rastreados é considerado (veja Figura 5.14(f)), obtém-se um espaço de votos bastante denso com um volume grande de dados espúrios, o que também leva a uma queda na qualidade do conjunto de votos estimados pelo RANSAC e, consequentemente, nos parâmetros da reta estimada pelo método dos mínimos quadrados. Em geral, os melhores resultados de nossa metodologia foram alcançados quando se considerou conjuntos de aproximadamente 4 objetos rastreados.

Conclusões e propostas de trabalhos futuros

Esta tese propõe uma metodologia inovadora para se alinhar no tempo e no espaço múltiplas sequências de vídeo adquiridas a partir de pontos de vista distintos. Especificamente, a abordagem apresentada neste trabalho reduz o problema de se estimar o alinhamento espaço-temporal entre sequências a dois subproblemas mais simples: um problema de regressão linear e um problema de otimização linear, enquanto todas os demais trabalhos relacionados ao nosso se baseiam na pesquisa da solução desejada em todo o espaço de alinhamentos possíveis, o que faz com que os mesmos possuam uma natureza combinatória.

A qualidade dos alinhamento estimados por nossa abordagem e seu correspondente custo computacional são invariantes à magnitude dos desalinhamentos temporais entre as sequências. Além disso, diferentemente dos demais métodos existentes na literatura, os quais são dedicados a conjuntos de apenas duas sequências de vídeo, nossa abordagem é capaz de alinhar num único passo um número arbitrário de sequências.

Os experimentos realizados nesta tese demonstraram que nossa abordagem é adequada para se resolver diversos problemas encontrados em aplicações atuais que se beneficiam da disponibilidade de registros simultâneos de um mesmo evento físico, fornecendo assim uma alternativa interessante às tecnologias de sincronização de vídeos baseadas em hardware, as quais são mais caras e mais difíceis de se utilizar em ambientes externos.

Do ponto de vista teórico, este trabalho demonstra sua relevância ao fornecer evidências adicionais que por meio da consideração de pistas temporais e espaciais em uma única metodologia, muitos eventos físicos que são inerentemente ambíguos para métodos tradicionais de alinhamento imagem-a-imagem são resolvidos eficazmente por técnicas de alinhamento sequência-a-sequência.

Como trabalhos futuros, pesquisas teóricas adicionais precisam ser consideradas. Primeiramente, deve-se estudar a criação de um modelo matemático alternativo para o desalinhamento temporal entre sequências de vídeo quando as mesmas são adquiridas a partir de câmeras que não trabalham a taxas de quadros constantes, um fato muitas vezes comum em aplicações de robótica. Além disso, técnicas alternativas para a estimação inicial da geometria epipolar de cada par de câmeras devem ser concebidas. Atualmente, nossa metodologia considera a existência de um conjunto suficiente de pontos estáticos na cena que sejam visíveis pelas câmeras, os quais possam ser utilizados pelo algoritmo dos oito pontos normalizado para se estimar a matriz fundamental. Sabe-se, entretanto, que este cenário pode não acontecer em alguns casos.

Finalmente, uma outra direção importante para pesquisa futura consiste na combinação de nossa metodologia com técnicas de reconstrução de cenas 3D para melhorar a eficácia de tais métodos. Atualmente, estamos pesquisando a extensão da abordagem proposta nesta tese para a realização do alinhamento espaço-temporal entre não somente câmeras baseadas no modelo de projeção em perspectiva como também câmeras catadióptricas.

To my parents, brothers and sister, who taught me to persevere and prepared me to face challenges with faith and humility. They are a constant source of inspiration to my life.

Acknowledgments

It's been quite a ride getting to this point, and I owe many thanks to several people for many things.

Special thanks to my advisor, Rodrigo Carceroni, for sharing with me his great ideas and for giving me the opportunity to get this far. Without his support, advice and close supervision, this thesis would not be possible.

Thanks a lot to Geraldo Massahud for his invaluable help in designing and building the system proposed in this thesis for aligning both in time and space multiple video sequences. It was really funny to work with someone who has an amazing ability to disagree to almost all conceivable points of view and a complex nonlinear way of thinking!

Thanks to Professor Kyros Kutulakos, who also contributed significantly for the development of this work. By working with him when writing our paper to CVPR, I could understand what Professor Carceroni meant with: "Professor Kyros has an unstoppable determination to reach perfection and an extensive knowledge of the rules of the game...".

I would like to thank the members of my thesis committee, professors Guilherme Pereira, Hani Yehia, Paulo Carvalho and Siome Goldenstein, for their suggestions to improve this work.

Many thanks to my colleagues and friends at VERLab, for the excellent work atmosphere and the valuable discussions during the development of this

thesis. In special, I would like to thank the support of Pedro Shiroma, Vilar Neto, Guilherme Pereira, Erikson Morais, Wagner Barros, José Pinheiro and Bruno Pimentel.

I would like to express my gratitude to professors Mario Campos and Antônio de Pádua Braga for showing me the beauty of a research career.

I would also like to thank my family for the support they provided me through my entire life and in particular, I must acknowledge my brothers Pedro and Paulo, who had a direct influence in my choice for assuming a scientific career.

Special thanks to my girlfriend and best friend, Fabiana, without whose love and encouragement, I would not have finished this thesis.

Thanks to the administrative staff of the Department of Computer Science at UFMG, for making things run smoothly.

I thank CNPq for the financial support.

Finally, I thank all people with whom I've worked and who participated directly or indirectly in this work.

Contents

List of Figures

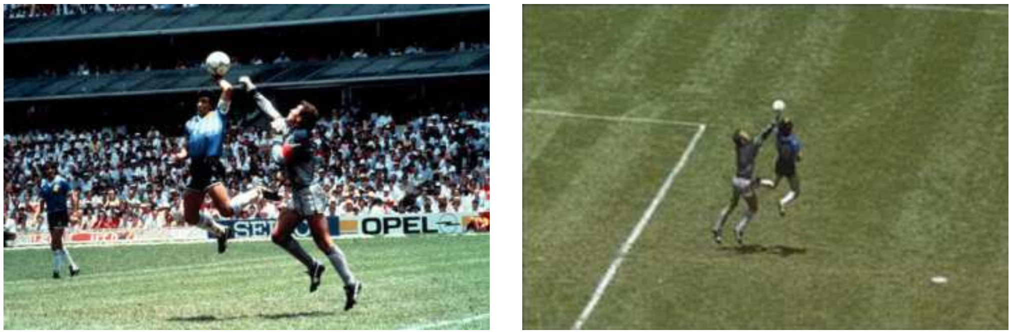
Chapter 1

Introduction

> I don't want to achieve immortality through my work.
> I want to achieve it through not dying.
>
> *Woody Allen*

In this work we consider the problem of *Spatio-Temporal Alignment of Multiple Video Sequences* of the same 3D scene, captured from distinct viewpoints. The scene dynamics as well as static scene features are used as powerful cues for estimating the *temporal synchronization* (temporal alignment) and the *spatial alignment* between the sequences. Typically, the *temporal misalignment* between video sequences arises from two main reasons. The first one relates to the fact that the input sequences may have different frame rates (e.g., NTSC and PAL), while the second one relates to the existence of a time shift or offset between the sequences, which is frequently created when the cameras are not activated simultaneously. On the other hand, the *spatial misalignment* results from the different positions, orientations and internal calibration parameters of all the cameras.

Real-world scenes where objects move and deform in 3D space are often so complex that, in order to understand them completely, it is necessary to observe them simultaneously from multiple viewpoints (Carceroni, 2001;



(a) First viewpoint. (b) Second viewpoint.

Figure 1.1: The famous *Hand of God goal* scored by Maradona in the 1986 FIFA World Cup match between England and Argentina (FIFA, 2004b).

Kutulakos, 2000; Kutulakos and Seitz, 2000; Szeliski, 1999). Consider, for instance, a sports event. Even a well-trained referee who is closely observing a scene of this type sometimes fails to capture pieces of information that are essential for accurate judgment. Figure 1.1 illustrates a famous example of such a failure, where two views of *The Hand of God goal* are presented (FIFA, 2004b). That goal was scored by Maradona, who knocked the ball into the net with the back of his left hand rather than with his head, a famous incident in the 1986 FIFA World Cup quarter-final match between England and Argentina in the Aztec Stadium, Mexico City.

In many of these situations, the missing visual clues are later revealed clearly in video sequences captured by strategically-positioned cameras, making the single-observer error evident. In particular, one of the major obstacles to be overcomed in the analysis of events such as the one in Figure 1.1 is represented by the need of a previous spatio-temporal alignment between the sequences, that is, the need of establishing correspondences in time and in space between the different image sequences.

The demand for effective automatic methods for temporally and spatially aligning multiple videos, mainly pre-recorded videos (e.g., regular and slow-motion clips of the same penalty kick), where video synchronization hardware and calibration techniques cannot be applied, has directed us in our quest to develop a novel solution that takes into account the constraints related to the monitored scene, as well as the constraints related to the set of cameras. Within this context, our main goal in this work consists in advancing towards the development of new practial methods for accuretely aligning both in time and space not only two sequences, as most of the existing methods, but a general set of N video sequences captured from distinct viewpoints. Therefore, the problem that we are addressing in this thesis can be stated as follows:

Given a dynamic scene, viewed simultaneously by N perspective cameras located at distinct viewpoints, which work with constant (albeit not necessarily identical) temporal sampling rates and constant (but unknown) intrinsic and extrinsic parameters, recover the N-dimensional function that captures the temporal relations between all sequences as well as the spatial transformations that align them in space.

Specifically, our work focuses on sets of video sequences that have overlap between their fields of view — *i.e.*, given N sequences $S_1, ..., S_N$, we can identify corresponding tuples of pixels $(x_1, y_1, t_1) \in S_1$,...,$(x_N, y_N, t_N) \in S_N$, where each such tuple is formed by projections of a single point $\mathbf{Q}$ in the scene space-time — in such a way that these overlapping parts contain some non-rigid motion.

1.1 Motivation

Many applications today benefit from the availability of simultaneous video recordings of the same physical event. Examples include tele-immersion (Vedula et al., 2002), video-based surveillance (Zelnik-Manor and Irani, 2001), video mosaicing (Caspi and Irani, 2001), video metrology from television broadcasts of athletic events (Reid and Zisserman, 1996) and recovering of the affine structure of a non-rigid motion (Tresadern and Reid, 2003). A critical task in all of these applications is the spatio-temporal alignment between the videos involved. Frequently, the use of special synchronization hardware has been the most common solution adopted to acquire temporally aligned sequences. However, alignment based on the content of the image sequences themselves has proved to be a more interesting alternative, since that it is less expensive, easier to use outside labs, and it can be applied to various multi-view sequences that already exist in video databases, such as those of sport events (FIFA, 2004a; Reid and Zisserman, 1996), artistic performances, or crimes captured in survelillance tapes.

In Reid and Zisserman (1996), for instance, the authors combined information from two independent sequences, illustrated in Figure 1.2, to resolve the controversy regarding the supposed goal scored by the English player Geoff Hurst in the 1966 FIFA World Cup. In particular, they wanted to know if the ball had actually crossed the goal line, and if not, how close it came to crossing it. In that work, the authors manually synchronized the sequences and then computed spatial alignment between selected corresponding images. This is an interesting example where automatic spatio-temporal alignment may provide better results.

In Caspi and Irani (2001), the authors illustrate an interesting application by performing the alignment of multi-sensor sequences for sensor fusion.

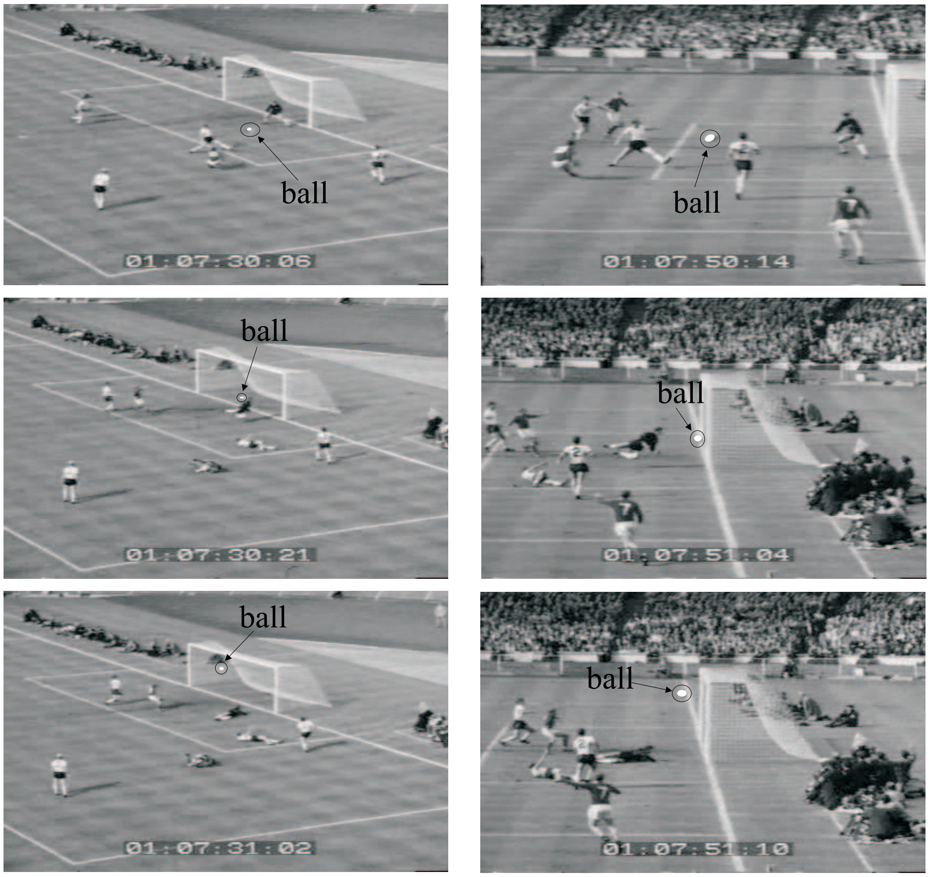
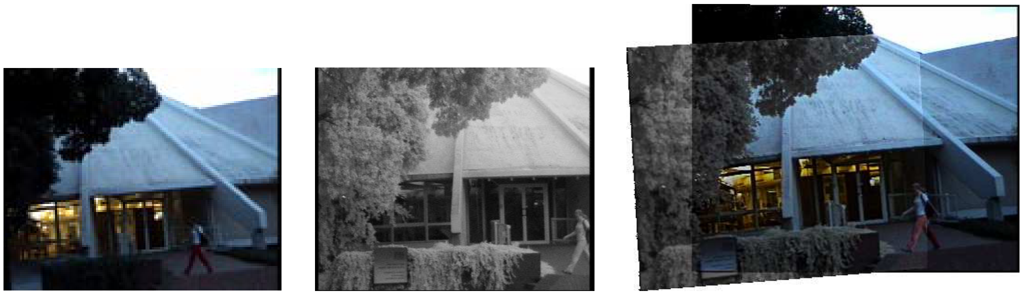


Figure 1.2: Images from two available sequences from the incident regarding the supposed goal scored by the English player Geoff Hurst in the 1966 FIFA World Cup (Reid and Zisserman, 1996).

Figure 1.3 shows the example presented by the authors. Two sequences of an outdoor scene were aligned, one captured by an Infra-Red camera, while the other by a regular video (visible-light) camera. The result of the spatio-temporal alignment is illustrated by fusing temporally corresponding frames. The Infra-Red camera provides only intensity information, and was therefore fused with the intensity component of the visible-light camera.

A final example of the importance of effective video alignment techniques

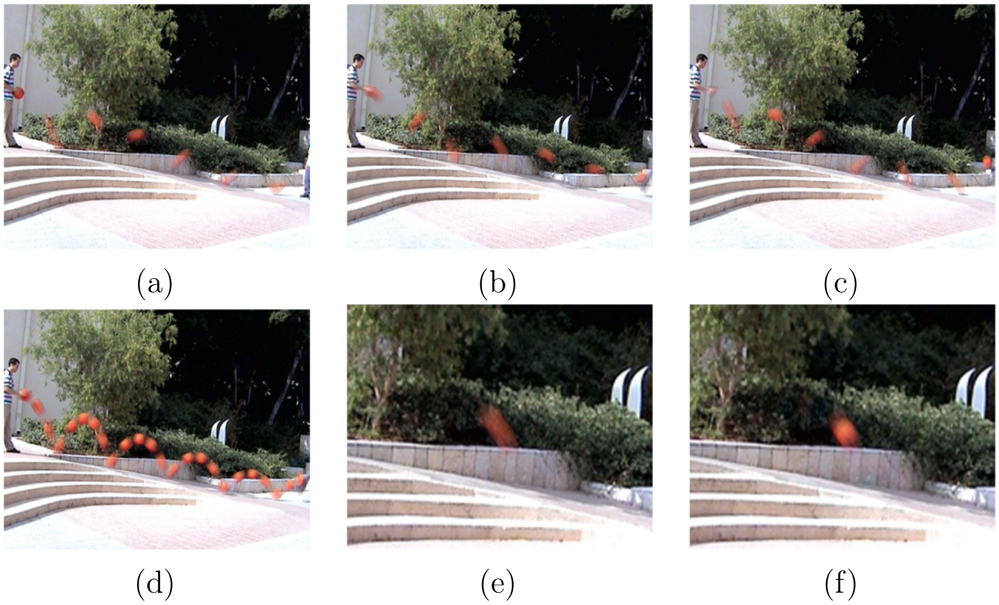


(a) Visible Light. (b) Infra-Red. (c) Output.

Figure 1.3: Example of application of spatio-temporal alignment of multiple sequences for multi-sensor fusion. (a) and (b) are temporally corresponding frames from the visible-light and Infra-Red sequences, respectively. (c) shows the results of fusing the two sequences after spatio-temporal alignment. (Caspi and Irani, 2001).

is given in Shechtman et al. (2002), where the authors propose a novel method for constructing a video sequence of high space-time resolution by combining information from multiple low-resolution video sequences of the same dynamic scene. Some results are illustrated in Figure 1.4. In this work, the spatio-temporal alignment method proposed in Caspi and Irani (2000) is applied in a preliminar step.

Importantly, content-based alignment of sequences acquired with stationary cameras is possible only if (a) these sequences depict parts of the scene space-time that have some overlap and (b) the regions visible in both sequences move in a way that is not completely rigid. If both conditions above hold, then the existence of a rigid transformation between the overlapping parts of any two frames (one from each sequence) indicates that these frames were probably acquired simultaneously. Every solution to the alignment problem that we address here (ours included) exploits this constraint.



(a) (b) (c)

(d) (e) (f)

Figure 1.4: Alignment and integration of information across multiple video sequences to exceed the limited spatial-resolution and limited temporal-resolution of video cameras. (a)-(c) Display the event captured by three low-resolution sequences. (d) The reconstructed event as captured by the generated high-resolution sequence. (e) A close-up image of the distorted ball in one of the low resolution frames. (f) A close-up image of the ball at the exact corresponding frame in time in the high-resolution output sequence (Shechtman et al., 2002).

1.2 Approach

We believe that any general solution to the spatio-temporal alignment problem should handle the following cases:

- **Unknown frame rate:** The relative frame rate of the video sequences is unknown and unconstrained.

- **Arbitrary time shift:** The time shift between the sequences is unknown and can be arbitrarily large.

- **Unknown motion:** The 3D motion of objects in the scene is unknown and unconstrained.
- **Tracking failures:** Individual scene points cannot be tracked reliably over many frames.
- **Unknown epipolar geometry:** The relative camera geometry of the video sequences is unknown.
- **Scalability:** Computational efficiency should degrade gracefully with increasing number of sequences.
- **No static points:** No visible point in the scene remains stationary for two or more frames.

As a step toward this goal, we present a novel approach that operates under all of the above conditions except the last one. In particular, we assume that for every pair of video sequences we can identify enough static scene points to get an initial estimate of the cameras' epipolar geometry. Moreover, in order to ensure that the parameters of that initial estimate remain constant during the application of our approach, we consider a scenario where the video cameras are stationary, with fixed (but *unknown*) intrinsic and extrinsic parameters.

Because of that last assumption, every corresponding set of N pixels is related by the same spatio-temporal transformation, whose spatial components are temporally invariant. Moreover, as long as each individual image in each sequence is acquired instantaneously, the temporal component of this transformation is spatially invariant, which allows us to decouple its recovery from the recovery of the spatial alignment. Importantly, our methodology still can be used in the case of moving cameras, as long as preliminary video stabilization process is applied.

We further assume that both cameras follow the projective pinhole model, and that they acquire frames at constant (albeit not necessarily identical) temporal sampling rates. The constant sampling rate assumption implies that the temporal coordinates (frame numbers) in one reference sequence and the temporal coordinates in all other sequences are related by a one-dimensional affine transformation:

$$f_i = \alpha_i f_r + \beta_i, \tag{1.1}$$

where f_i and f_r are the frame numbers of the i-th sequence and the reference sequence, respectively, and α_i, β_i are unkown constants describing the temporal dilation and temporal shift, respectively, between the sequences. In general, these constants will not be integers.

The pairwise temporal relations captured by Equation (1.1) induce a global relationship between the frame numbers of the input sequences. In fact, at the heart of our approach lies the concept of a *timeline*. Given N sequences, the timeline is a N-dimensional line that completely describes all temporal relations between the sequences. A key property of the timeline is that even though knowledge of the timeline implies knowledge of the sequences' temporal alignment, we can compute points on the timeline without knowing this alignment. Using this property as a starting point, we reduce the temporal alignment problem for N sequences to the problem of robustly estimating a single N-dimensional line from a set of appropriately-generated points in $\mathbb{R}^N$.

Finally, we assume in this work that the overlapping sequence parts contain features such as uniformly-colored blobs or corners that move along (piecewise) smooth trajectories, and these trajectories can be captured by trackers that output trajectory segments detected in all the sequences as parametric curves. We do not assume that correspondences between trajectories are known *a priori*: recovering such correspondences is part of the job done by the method that we will introduce in the next chapters. Therefore, in practice the data that we need can be obtained with one of the many single-view, real-time, multi-feature trackers available in the literature (Lowe, 2004; Jepson et al., 2003; Isard and MacCormick, 2001; Shi and Tomasi, 1994).

1.3 Contributions

From a practical standpoint, this work has ushered in two major contributions to the area of video analysis, specially, to the field of applications where multiple video sequences must be temporally and spatially aligned:

- A generalized framework for solving the problem of spatio-temporal alignment between N videos sequences captured from distinct view points. The framework (1) can handle arbitrary large misalignments between the sequences, (2) does not require any *a priori* information about their temporal relations, (3) does not assume that a single scene point can be tracked reliably over the entire sequence, (4) does not require the ability to establish feature correspondences between the sequences, (5) can handle sequences with feature trajectories that nearly overlap in 3D, that are approximately cyclical and that contain features with quite large image velocities (up to 9 pixels per frame), (6) can handle sequences with multiple frame rates and that contain spurious clips, and (7) can handle sequences where the tracked features move in completely distinct planes in the scene.

- A new iterative algorithm to refine simultaneously the parameters representing the temporal and spatial relations between the sequences, since that the exclusive refinement of the temporal parameters is suboptimal.

From a theoretical point-of-view, this work is important because it provides additional theoretical and empirical evidence that by considering temporal and spatial cues into a single alignment framework, many events that are inherently ambiguous for traditional image-to-image aligment techniques can be uniquely resolved by sequence-to-sequence alignment methods.

1.4 Outline of this work

This document is organized in six chapters (including this one) and three appendices. Chapter 2 discusses the state-of-the-art in Spatio-Temporal Alignment of Multiple Video Sequences and introduces some fundamental concepts in single and multiple view geometries that we use as a theoretical basis to our work. In Chapter 3, we present the *timeline constraint* and our temporal alignment methodology associated to that concept. Chapter 4 presents an iterative algorithm for refining simultaneously the parameters representing the temporal and spatial relations between the sequences. In Chapter 5, we present and discuss experimental results with real-world and synthetic video sequences. Chapter 6 presents our conclusions and perspectives of future work. Appendix A describes in detail the RANSAC algorithm. Appendix B gives a brief introduction on tensorial notation. Finally, Appendix C describes an important tool applied in the analysis of three-view geometries: the *trifocal tensor*, and some techniques for deriving multi-linear constraints on correspondences in the case of N-views.

Chapter 2

Background

> Everything has been said before, but since nobody listens we have to keep going back and beginning all over again.
>
> *Andre Gide*

This chapter presents a survey of the current state-of-the-art in Spatio-Temporal Alignment of Multiple Video Sequences and introduces some fundamental concepts in single and multiple view geometries, specially two-view geometries, that we use as a theoretical basis to our work. Related work that support specific topics of this thesis will be surveyed wherever necessary.

2.1 Spatio-Temporal Alignment

In spite of the fact that spatial alignment is one of the problems that have been most heavily researched by the Computer Vision community, so far only a handful of works have dealt with the problem of temporal alignment.

Based on the analysis of the most relevant articles found in the literature, we classify the existing spatio-temporal alignment methods in two main groups: the *feature–based* methods and the *direct* methods. The feature–

based methods (Caspi et al., 2002; Rao et al., 2003; Wolf and Zomet, 2002a,b; Lee et al., 2000; Stein, 1998) extract all information needed to perform spatio-temporal alignment from the trajectories of tracked features, such as uniformly–colored blobs or corners. On the other hand, direct methods (Caspi and Irani, 2000, 2001) extract that information from the intensities and intensity gradients of all pixels that belong to overlapping regions. Therefore, direct methods tend to align sequences more accurately if their appearances are similar, while feature–based methods are widely prescribed (Caspi et al., 2002; Rao et al., 2003; Torr and Zisserman, 1999) for sequences with dissimilar appearance such as those acquired with wide baselines, different magnifications, or by cameras with distinct spectral sensitivities.

In this work, we propose a novel feature–based methodology for sequence–to–sequence alignment. More specifically, a major novelty of our methodology is that it reduces the computation of temporal and spatio-temporal alignments between sequences to linear regression and linear optimization problems, while existing feature–based techniques (Caspi et al., 2002; Rao et al., 2003; Wolf and Zomet, 2002a,b; Lee et al., 2000; Stein, 1998) search the entire space of possible temporal alignments. Unfortunately, the combinatorial nature of this search requires several additional assumptions to make it manageable, such as (1) known cameras' frame rates, (2) the number of video sequences N is restricted to be two, (3) temporal misalignment as an integer, and (4) temporal misalignment within a small user-specified range (typically less than fifty frames).

Unlike previous feature–based techniques, our approach aligns N sequences in a single step. It can handle arbitrarily large misalignments between them and does not require any *a priori* information about their temporal relations.

The quality of the alignments that we obtain and the computational cost of our alignment process are invariant to the magnitude of the initial temporal offsets between sequences. To achieve this breakthrough, we derive alignment constraints by matching instantaneous positions of features in one sequence against entire feature trajectories in the other sequences, while most feature–based techniques rely on matches between pairs of instantaneous positions.

Our key observation is that, because feature trajectories are (piecewise–) smooth curves parameterized by time, once the geometric relations of a set of cameras can be estimated, each match between an instantaneous feature in one reference sequence and a trajectory in one of the other sequences constrains the feature's temporal coordinate to be aligned with one among a finite set of instants in the other sequence: those instants where the feature's epipolar line (see the concept of epipolar line in Section 2.3.1) intersects the matching trajectory. Importantly, these instants where intersections occur not necessarily correspond to the second sequence's frames, which means that our methodology yields temporal alignment at sub–frame accuracy, contrary to techniques based on position–to–position matches.

In this thesis, we exploit the observation above to develop a sequence–to–sequence alignment approach based on two techniques: (a) one that builds large sets of those temporal constraints from a rough spatial alignment between sequences and then performs a robust linear regression in the temporal domain to recover the globally correct temporal alignment, and (b) one that linearizes feature trajectories around the points of intersection with epipolar lines to reduce the problem of finding the complete spatio–temporal alignment between two sequences to a problem of solving a linear system.

Our work is most closely related to the approach of Caspi, Simakov and Irani (Caspi et al., 2002). In their approach, the epipolar geometry and temporal misalignment between two sequences are recovered from the image trajectory of a single scene point that is visible in both sequences, and are subsequently refined using more features. To achieve this, they assume known frame rates and formulate a non-linear optimization problem to jointly estimate epipolar geometry and temporal misalignment. Unfortunately, the highly non-linear nature of this optimization necessitates good initial estimates for the temporal misalignment and the epipolar geometry.

Importantly, that approach still assumes that a single scene point can be tracked reliably over the entire sequence. This may be difficult to achieve when aligning videos of complex scenes, where feature tracking can fail often because of occlusions or large inter-frame motions. Our solution, on the other hand, requires the ability to track scene points only across two consecutive frames of the same sequence. Moreover, it does not require the ability to establish feature correspondences between the sequences.

All other feature–based methods for spatio–temporal alignment that we are aware of (Rao et al., 2003; Wolf and Zomet, 2002a,b; Lee et al., 2000; Stein, 1998) use position–to–position constraints. It is known (Hartley and Zisserman, 2003; Ma et al., 2003) that in the case of two-view geometries — except in degenerate cases — a match between static points in images acquired by two stationary cameras generates a linear constraint on the parameters of the cameras' fundamental matrix. It is also known (Hartley and Zisserman, 2003; Ma et al., 2003) that in static scenes (or in cases where two sequences are correctly synchronized) the matrix formed by all such constraints is singular. The methods cited above use this fact to check if any possible temporal transformation between two sequences is consistent with the instantaneous feature positions in the sequences. They do this by assembling large constraint matrices from multiple position–to–position matches and then testing these matrices for rank–deficiency.

Thus, all those methods need to search the space of possible temporal transformations to find the optimum alignment. Rao et al. (Rao et al., 2003) have recently proposed a way to perform this search in an incremental, non-exaustive way, but their solution assumes that the initial frames of the two sequences are aligned *a priori*. Moreover, because of this search–based nature, none of the techniques based on position–to–position constraints can synchronize sequences at sub–frame accuracy and only one of them (Wolf and Zomet, 2002b) can tolerate outliers in the matched position pairs, albeit by assuming that both cameras are orthographic.

Finally, there is the method proposed by Caspi and Irani (Caspi and Irani, 2000) that aligns sequences directly from pixel intensities and their spatio–temporal gradients. Because it uses only linear terms of intensities' Taylor–series expansions to approximate spatial and temporal intensity variations that are in general non–linear, it only works if the initial sequence–to–sequence misalignments in space and time are small enough to fall within the range of validity of intensity linearizations. In addition, it models spatial transformations between sequences as homographies, which — contrary to fundamental matrices — are not appropriate to align sequences with signifficant depth discontinuities. Homographies and fundamental matrices are geometric relations between two views, which are introduced in Section 2.3.1.

Caspi and Irani also developed a direct method that can align sequences without any overlap (Caspi and Irani, 2001), but this later method does not work with stationary cameras: it only works with sequences acquired by pairs of cameras that remain rigidly attached to each other while moving relative to a mostly rigid scene.

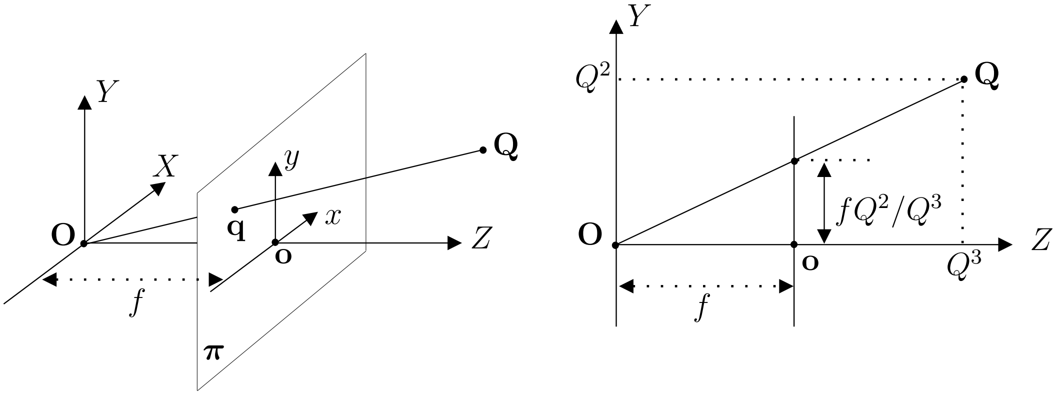


Figure 2.1: The pinhole camera model (Hartley and Zisserman, 2003).

2.2 Single View Geometry

This section concentrates on the introduction of some fundamental concepts in the geometry of a single perspective camera. Basically, a camera is a mapping between the 3D world (object space) and a 2D image (Trucco and Verri, 1998). There are several camera models in the literature, which are matrices with particular properties that represent the camera mapping. Specifically, we consider in this work that all cameras follow the *projective pinhole model*, which is illustrated in Figure 2.1. The basic pinhole model consists of a plane $\boldsymbol{\pi}$, the *image plane*, and a 3D point $\mathbf{O}$, the *center of projection*. The distance f between $\boldsymbol{\pi}$ and $\mathbf{O}$ is the *focal length*. The line through $\mathbf{O}$ and perpendicular to $\boldsymbol{\pi}$ is the *optical axis*, and $\mathbf{o}$, the intersection between $\boldsymbol{\pi}$ and the optical axis, is named *image center*. As illustrated in Figure 2.1, $\mathbf{q}$, the image of $\mathbf{Q}$, is the point at which the straight line through $\mathbf{Q}$ and $\mathbf{O}$ intersects the image plane $\boldsymbol{\pi}$. Consider the 3D reference frame in which $\mathbf{O}$ is the origin and the plane $\boldsymbol{\pi}$ is orthogonal to the Z axis, and let $\mathbf{Q} = (Q^1, Q^2, Q^3)^\top$. By similar triangles, one quickly computes that the point $\mathbf{Q}$ is mapped to the point $\mathbf{q} = (fQ^1/Q^3, fQ^2/Q^3, f)^\top$ on the image plane. Ignoring the final image coordinate, we see that

$$(Q^1, Q^2, Q^3)^\top \mapsto (fQ^1/Q^3, fQ^2/Q^3)^\top \tag{2.1}$$

describes the *central projection* mapping from world to image coordinates (Hartley and Zisserman, 2003). This is a mapping from Euclidean 3D-space $\mathbb{R}^3$ to Euclidean 2D-space $\mathbb{R}^2$.

If world and image points are represented by homogeneous vectors, then the central projection is expressed as a linear mapping between their homogeneous coordinates (Hartley and Zisserman, 2003). In particular, Equation (2.1) may be written in terms of matrix multiplication as

$$\begin{bmatrix} Q^1 \\ Q^2 \\ Q^3 \\ 1 \end{bmatrix} \mapsto \begin{bmatrix} fQ^1 \\ fQ^2 \\ Q^3 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} Q^1 \\ Q^2 \\ Q^3 \\ 1 \end{bmatrix}. \tag{2.2}$$

Computer vision algorithms for reconstructing the 3D structure of a scene or computing the position of objects in space need equations as Equation (2.2) for linking the coordinates of points in 3D space with the coordinates of their corresponding image points (Trucco and Verri, 1998). In these applications it is often assumed that the coordinates of the image points in the camera reference frame can be obtained from *pixel coordinates* and that the camera reference frame can be located with respect to some other, known as the *world reference frame* (Trucco and Verri, 1998). This is equivalent to assume knowledge of some camera's characteristics, known in vision as the camera's *extrinsic* and *intrinsic* parameters. In the following we briefly introduce the definitions of extrinsic and intrinsic parameters in practical terms. The problem of estimating their values is called *camera calibration* (Hartley and Zisserman, 2003).

2.2.1 Extrinsic parameters

The extrinsic parameters are defined as any set of geometric parameters that identify uniquely the transformation between the unknown camera reference frame and a known reference frame, named the *world reference frame* (Trucco and Verri, 1998).

A typical choice for describing the transformation between camera and world frame is to use (Trucco and Verri, 1998; Hartley and Zisserman, 2003):

- a 3 × 1 translation vector $\mathbf{t}$, describing the relative positions of the origins of the two reference frames, and
- a 3 × 3 rotation matrix $\mathcal{R}$, an orthogonal matrix that brings the corresponding axes of the two frames onto each other.

The relation between the coordinates of a point $\mathbf{Q}$ in world and camera frame, $\mathbf{Q}_w$ and $\mathbf{Q}_c$ respectively, is

$$\mathbf{Q}_c = \mathcal{R}\mathbf{Q}_w + \mathbf{t}. \tag{2.3}$$

Now, if we add a "1" as a fourth coordinate of $\mathbf{Q}_w$ (that is, express $\mathbf{Q}_w$ in homogeneous coordinates), we may group the extrinsic parameters in a single 3 × 4 matrix $\mathcal{M}_e$, called *matrix of extrinsic parameters*

$$\mathcal{M}_e = \begin{bmatrix} \mathcal{R} & \mathbf{t} \end{bmatrix}, \tag{2.4}$$

and obtain the following linear matrix equation for linking the coordinates of points in the world reference frame with their corresponding coordinates in the camera reference frame

$$\mathbf{Q}_c = \mathcal{M}_e \begin{bmatrix} \mathbf{Q}_w \\ 1 \end{bmatrix}. \tag{2.5}$$

2.2.2 Intrinsic parameters

The intrinsic parameters can be defined as the set of parameters needed to characterize the optical, geometric, and digital characteristics of the viewing camera (Trucco and Verri, 1998). They are the focal length f, the location of the image center in pixel coordinates (o_x, o_y), the effective pixel size in the horizontal and vertical direction (s_x, s_y), and, if required, a radial distortion coefficient k.

Neglecting radial distortion, we can group all the intrinsic parameters in a single 3×3 matrix $\mathcal{M}_i$, called *matrix of intrinsic parameters*

$$\mathcal{M}_i = \begin{bmatrix} -f/s_x & 0 & o_x \\ 0 & -f/s_y & o_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.6}$$

Again, if we express $\mathbf{Q}_w$ in homogeneous coordinates, we obtain the following linear matrix equation describing perspective projections

$$\mathbf{q} = \mathcal{M}_i \mathcal{M}_e \begin{bmatrix} \mathbf{Q}_w \\ 1 \end{bmatrix}. \tag{2.7}$$

What is interesting about vector $\mathbf{q} = [q^1, q^2, q^3]^\top$ is that the ratios (q^1/q^3) and (q^2/q^3) are nothing but the coordinates in pixel of the image point. Therefore, $\mathcal{M}_i$ performs the transformation between the camera reference frame and the image reference frame.

2.3 Multiple View Geometry

A basic problem in computer vision is to understand the structure of a real world scene given several images of it (Hartley and Zisserman, 2003). Over the past decade there has been a rapid development in the understanding and modelling of the geometry of multiples views, especially due to the new achievements in our theoretical understanding and improvements in the estimation of mathematical objects from images (Hartley and Zisserman, 2003; Ma et al., 2003; Forsyth and Ponce, 2002).

While several problems of scene reconstruction have already reached reasonable solutions, such as the problem of estimating a multifocal tensor from image point correspondences, particularly the fundamental matrix and the trifocal tensor (Hartley and Zisserman, 2003), other relevant problems still claim for more carefull study. Examples include: (1) application of bundle adjustment to solve more general reconstruction problems, and (2) automatic detection of correspondences in image sequences with elimination of outliers and false matches using the multifocal tensor relationships.

Next, we will briefly describe some important concepts and definitions in two-view geometry, which constitute the basis of our spatio-temporal alignment methodology, such as the epipolar geometry of two cameras. In Appendix C, we present more general frameworks that are natural extensions for three-, four- and N-views. Particularly, we introduce in this appendix the trifocal tensor, which plays an analogous role in three views to that played by the fundamental matrix in two. Finally, we still describe some techniques for deriving multi-linear constraints on correspondences in the case of N-views.

2.3.1 Two-View Geometry

The geometry of two perspective views may be acquired simultaneously as in a stereo rig, or acquired sequentially, for example by a camera moving relative to the scene. We say that these two situations are geometrically equivalent and they will not be differentiated here. Basically, there are two relations between two views of a scene (Hartley and Zisserman, 2003; Ma et al., 2003):

1. The *homography* of a point in one view determines a point in the other which is the image of the intersection of the ray with a plane, as illustrated in Figure 2.2.

2. The *epipolar geometry* a point in one view determines a line in the other which is the image of the ray through that point, as illustrated in Figure 2.3.

In the following we briefly introduce these both relations.

The homography map

Images of points on a plane are related to corresponding image points in a second view by a (planar) *homography* as shown in Figure 2.2. This is a projective relation since it depends only on the intersections of planes with lines (Forsyth and Ponce, 2002). We say that the world plane induces a homography between the views and that the homography map is responsible for transferring points from one view to the other.

In Hartley and Zisserman (2003), it is shown that for world planes in *general position* the homography is determined uniquely by the plane and vice versa. In this case, general position means that the world plane does not contain either of the camera centers. If the world plane does contain one

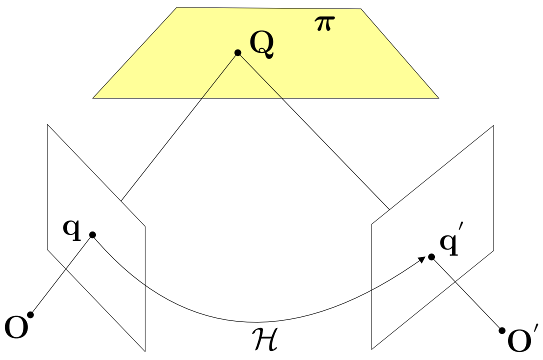


Figure 2.2: The homography map induced by a world plane (Hartley and Zisserman, 2003). The ray corresponding to a point $\mathbf{q}$ is extended to meet the world plane $\boldsymbol{\pi}$ in a point $\mathbf{Q}$. This point is projected to a point $\mathbf{q}^{'}$ in the other image. The map from $\mathbf{q}$ to $\mathbf{q}^{'}$ is the *homography* induced by the plane $\boldsymbol{\pi}$. If $\mathcal{H}_1$ and $\mathcal{H}_2$ are the perspectivities from the world plane $\boldsymbol{\pi}$ to the first and second image planes, respectively, we have $\mathbf{q} = \mathcal{H}_1\mathbf{Q}$ and $\mathbf{q}^{'} = \mathcal{H}_2\mathbf{Q}$. It is the composition of these two perspectivities that defines a homography $\mathcal{H}$, $\mathbf{q}^{'} = \mathcal{H}_2{\mathcal{H}_1}^{-1}\mathbf{q} = \mathcal{H}\mathbf{q}$, between the image planes.

of the camera centers then the induced homography is degenerate (Hartley and Zisserman, 2003; Ma et al., 2003; Forsyth and Ponce, 2002).

In the following we derive an explicit expression for the induced homography between the two views. Suppose a world plane $\boldsymbol{\pi}$ as the one illustrated in Figure 2.2, which is specified by its coordinates in the world frame. Consider the following projection matrices for the two views

$$\mathcal{M} = \left[\begin{array}{c|c} \mathcal{I} & \mathbf{0} \end{array}\right] \qquad \mathcal{M}^{'} = \left[\begin{array}{c|c} \mathcal{A} & \mathbf{a} \end{array}\right]$$

where $\mathcal{I}$ is a 3×3 identity matrix, $\mathbf{0} = [0, 0, 0]^\top$ is a null 3-vector and $\mathcal{A}$ and $\mathbf{a}$ are the parameters of the projection matrix $\mathcal{M}^{'}$. Moreover, let $\boldsymbol{\pi}$ be a world plane with $\boldsymbol{\pi} = [\mathbf{v}^\top, 1]^\top$. Then the homography induced by the plane is $\mathbf{q}^{'} = \mathcal{H}\mathbf{q}$ with

$$\mathcal{H} = \mathcal{A} - \mathbf{a}\mathbf{v}^{\top}. \tag{2.8}$$

We may assume the fourth coordinate of $\boldsymbol{\pi}$ equal to 1 since the plane does not pass through the center of the first camera at $[0, 0, 0, 1]^{\top}$ (Hartley and Zisserman, 2003).

Observe that there is a three-parameter family of planes in the 3D world, and correspondingly a three-parameter family of homographies between two views induced by planes in the 3D world. These three parameters are specified by the elements of the vector $\mathbf{v}$, which is not a homogeneous 3-vector (Hartley and Zisserman, 2003).

Epipolar Geometry

The epipolar geometry between two views is essentially the geometry of the intersection of the image planes with the pencil of planes having the *baseline* as axis, where the baseline is the line joining the camera centers (Hartley and Zisserman, 2003; Ma et al., 2003; Trucco and Verri, 1998). The epipolar geometry is independent of scene structure, and only depends on the cameras' internal parameters and relative pose (Trucco and Verri, 1998).

The geometric entities involved in epipolar geometry are illustrated in Figure 2.3. This figure shows two pinhole cameras, their projection centers, $\mathbf{O}^{'}$ and $\mathbf{O}^{''}$, and image planes, $\boldsymbol{\pi}^{'}$ and $\boldsymbol{\pi}^{''}$. As usual, each camera identifies a 3D reference frame, the origin of which coincides with the projection center, and the Z-axis with the optical axis. The vectors $\mathbf{Q}^{'}$ and $\mathbf{Q}^{''}$ refer to the scene point, while the vectors $\mathbf{q}^{'}$ and $\mathbf{q}^{''}$ refer to the projections of the scene point onto the image planes $\boldsymbol{\pi}^{'}$ and $\boldsymbol{\pi}^{''}$, respectively, and are expressed in the corresponding reference frame. We call *epipole* the point of intersection of the baseline with the image plane. In Figure 2.3, we denote the epipoles

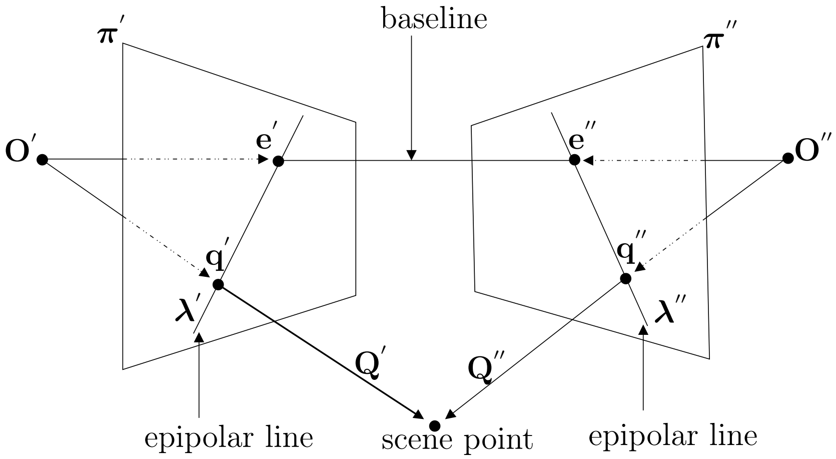


Figure 2.3: The epipolar geometry.

by $\mathbf{e}'$ and $\mathbf{e}''$. The plane identified by the scene point and the camera centers $\mathbf{O}'$ and $\mathbf{O}''$ is called *epipolar plane*, while its intersections $\boldsymbol{\lambda}'$ and $\boldsymbol{\lambda}''$ with the image planes $\boldsymbol{\pi}'$ and $\boldsymbol{\pi}''$, respectively, are called *epipolar lines.* With the exception of the epipoles, only one epipolar line goes through any image point.

The reference frames of both cameras in Figure 2.3 are related via the extrinsic parameters. Therefore, given a point in space, the relation between its representations $\mathbf{Q}'$ and $\mathbf{Q}''$ in the camera reference frames is

$$\mathbf{Q}'' = \mathcal{R}\left(\mathbf{Q}' - \mathbf{t}\right), \tag{2.9}$$

where $\mathbf{t} = (\mathbf{O}'' - \mathbf{O}')$ is the translation vector and $\mathcal{R}$ is the rotation matrix.

The relation between the scene point and its projections is described by the usual equations of perspective projection, in vector form:

$$\mathbf{q}' = \frac{f'}{Z'}\mathbf{Q}', \tag{2.10}$$

$$\mathbf{q}^{''} = \frac{f^{''}}{Z^{''}}\mathbf{Q}^{''}, \tag{2.11}$$

where $f^{'}$, $f^{''}$ are the cameras' focal lengths and $Z^{'}$, $Z^{''}$ are the coordinates in the Z-axis of $\mathbf{Q}^{'}$ and $\mathbf{Q}^{''}$, respectively.

Finally, another important concept is the so-called *epipolar constraint*. Consider the triplet formed by the scene point and their projections $\mathbf{q}^{'}$ and $\mathbf{q}^{''}$. Given $\mathbf{q}^{'}$, the scene point can lie anywhere on the ray from $\mathbf{O}^{'}$ through $\mathbf{q}^{'}$. However, since the image of this ray in $\boldsymbol{\pi}^{''}$ is the epipolar line through the corresponding point, $\mathbf{q}^{''}$, the correct match must lie on the epipolar line. This fact is known as epipolar constraint and establishes a mapping between points in $\boldsymbol{\pi}^{'}$ with lines in $\boldsymbol{\pi}^{''}$ and *vice versa*.

The Fundamental Matrix $\mathcal{F}$

The *fundamental matrix* is an algebraic representation of the epipolar geometry (Hartley and Zisserman, 2003; Ma et al., 2003). It is a basic tool in the development of our technique for spatio-temporal alignment between video sequences. In the following, in order to derive the fundamental matrix, we firstly derive another important matrix known as *essential matrix* (Longuet-Higgins, 1981).

The equation of the epipolar plane through the scene point can be written as the coplanarity condition of the vectors $\mathbf{Q}^{'}$, $\mathbf{t}$, and $\mathbf{Q}^{'} - \mathbf{t}$, or

$$\left(\mathbf{Q}^{'} - \mathbf{t}\right)^{\top} \mathbf{t} \times \mathbf{Q}^{'} = 0. \tag{2.12}$$

By using Equation (2.9), we obtain

$$\left(\mathcal{R}^{\top}\mathbf{Q}^{''}\right)^{\top} \mathbf{t} \times \mathbf{Q}^{'} = 0. \tag{2.13}$$

Given that a vector product can be rewritten as a multiplication by a rank-deficient matrix, we can write

$$\left(\mathbf{t} \times \mathbf{Q}^{'}\right) = \mathcal{J}\mathbf{Q}^{'}, \tag{2.14}$$

where

$$\mathcal{J} = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}. \tag{2.15}$$

By using this fact, Equation (2.13) becomes

$$\mathbf{Q}^{''\top}\mathcal{E}\mathbf{Q}^{'} = 0, \tag{2.16}$$

with

$$\mathcal{E} = \mathcal{R}\mathcal{J}. \tag{2.17}$$

The matrix $\mathcal{E}$ is known as the essential matrix (Longuet-Higgins, 1981) and establishes a link between the epipolar constraint and the extrinsic parameters of the stereo system. Observe that, by using Equations (2.10) and (2.11), and dividing by the product $Z^{'}Z^{''}$, Equation (2.16) can be rewritten as

$$\mathbf{q}^{''\top}\mathcal{E}\mathbf{q}^{'} = 0. \tag{2.18}$$

Consider now the matrices $\mathcal{M}_i^{'}$ and $\mathcal{M}_i^{''}$ of the intrinsic parameters of the cameras in Figure 2.3. Let $\overline{\mathbf{q}}^{'}$ and $\overline{\mathbf{q}}^{''}$ be the points in *pixel* coordinates corresponding to $\mathbf{q}^{'}$ and $\mathbf{q}^{''}$ in *camera* coodinates, that is

$$\overline{\mathbf{q}}^{'} = \mathcal{M}_i^{'}\mathbf{q}^{'}. \tag{2.19}$$

$$\overline{\mathbf{q}}^{''} = \mathcal{M}_i^{''} \mathbf{q}^{''}. \tag{2.20}$$

By substituting Equations (2.19) and (2.20) into Equation (2.18), we have

$$\overline{\mathbf{q}}^{''\top} \mathcal{F} \overline{\mathbf{q}}^{'} = 0, \tag{2.21}$$

where

$$\mathcal{F} = \mathcal{M}_i^{''-1} \mathcal{E} \mathcal{M}_i^{'-1}. \tag{2.22}$$

$\mathcal{F}$ is the so-called *fundamental matrix.* Observe that $\mathcal{F}\overline{\mathbf{q}}^{'}$ in Equation (2.21) can be thought of as the equation of the projective epipolar line, $\overline{\boldsymbol{\lambda}}^{''}$, that corresponds to point $\overline{\mathbf{q}}^{'}$, or

$$\overline{\boldsymbol{\lambda}}^{''} = \mathcal{F}\, \overline{\mathbf{q}}^{'}. \tag{2.23}$$

Observe that if it is possible to estimate the fundamental matrix from a number of point matches in pixel coordinates, we can reconstruct the epipolar geometry with no information at all on the intrinsic or extrinsic parameters.

We present below some of the most important properties of the fundamental matrix (Hartley and Zisserman, 2003):

- if $\mathcal{F}$ is the fundamental matrix of the pair of cameras $(\boldsymbol{\pi}^{'}, \boldsymbol{\pi}^{''})$, then $\mathcal{F}^\top$ is the fundamental matrix of the pair in the opposite order: $(\boldsymbol{\pi}^{''}, \boldsymbol{\pi}^{'})$.

- for any point $\mathbf{q}^{'}$ in the first image, the corresponding epipolar line is $\boldsymbol{\lambda}^{''} = \mathcal{F}\mathbf{q}^{'}$. Similarly, $\boldsymbol{\lambda}^{'} = \mathcal{F}^\top \mathbf{q}^{''}$ represents the epipolar line corresponding to $\mathbf{q}^{''}$ in the second image.

- for any point $\mathbf{q}^{'}$ (other than $\mathbf{e}^{'}$) the epipolar line $\boldsymbol{\lambda}^{''} = \mathcal{F}\mathbf{q}^{'}$ contains the epipole $\mathbf{e}^{''}$. Therefore, $\mathbf{e}^{''}$ satisfies $\mathbf{e}^{''\top}\left(\mathcal{F}\mathbf{q}^{'}\right) = (\mathbf{e}^{''\top}\mathcal{F})\mathbf{q}^{'} = 0$ for all $\mathbf{q}^{'}$. It follows that $\mathbf{e}^{''\top}\mathcal{F} = 0$ and, similarly, $\mathcal{F}\mathbf{e}^{'} = 0$.

- $\mathcal{F}$ is a rank 2 homogeneous matrix with seven degrees of freedom.

- $\mathcal{F}$ is a correlation, a projective map taking a point to a line. However, $\mathcal{F}$ is not a proper correlatiom, that is, $\mathcal{F}$ is not invertible.

Several methods of computing the fundamental matrix were presented in the literature (Zhang, 1998; Hartley and Zisserman, 2003) and many researchers still work on the development of new techniques. Of the various current methods, the *eight-point algorithm* (Hartley, 1997a) is by far the simplest. The idea behind this technique is based on the solution of a homogeneous linear system. Given a set of n point correspondences between two images, the fundamental matrix $\mathcal{F}$ satisfies the condition $\mathbf{q}_i^{''\top} \mathcal{F} \mathbf{q}_i^{'} = 0$, for $i = 1, ..., n$. With the $\mathbf{q}_i^{'}$ and $\mathbf{q}_i^{''}$ known, this equation is linear in the entries of the matrix $\mathcal{F}$. Thus, given at least 8 point correspondences it is possible to solve linearly for the entries of $\mathcal{F}$ up to scale. With more than 8 equations a least-squares solution is found. More detailed studies and caracterizations of the best methods for computing the fundamental matrix can be found in Hartley and Zisserman (2003) and Zhang (1998).

Chapter 3

Temporal Alignment

> O tempo propõe outras dificuldades. Uma, talvez a maior, a de sincronizar o tempo individual de cada pessoa com o tempo geral das matemáticas, foi fartamente apregoada pelo recente alarme relativista, e todos a recordam - ou lembram tê-la recordado até bem pouco tempo.
>
> *Jorge Luis Borges*

This chapter presents our framework for temporally aligning multiple sequences acquired from distinct viewpoints. We begin this chapter describing a key concept in our method: the *timeline*, an N-dimensional line responsible for capturing all the temporal relations between the video sequences.

3.1 The Timeline Constraint

Suppose that a dynamic scene is viewed simultaneously by N perspective cameras located at distinct viewpoints. We assume that each camera captures frames with a constant, unknown frame rate. We also assume that the cameras are unsynchronized, i.e., they began capturing frames at different times with possibly-distinct frame rates. In order to temporally align the resulting sequences, we must determine the correspondence between frame numbers in one "reference" sequence and frame numbers in all other sequences. This correspondence can be expressed as a set of linear equations,

$$f_i = \alpha_i f_r + \beta_i, \tag{3.1}$$

where f_i and f_r denote the frame numbers of the i-th sequence and the reference sequence, respectively, and α_i, β_i are unkown constants describing the temporal dilation and temporal shift, respectively, between the sequences. In general, these constants will not be integers.

The pairwise temporal relations captured by Equation (3.1) induce a global relationship between the frame numbers of the sequences. We represent this relationship by an N-dimensional line $\mathcal{L}$ that we call the *timeline*:

$$\mathcal{L} = \left\{ [\alpha_1 \ldots \alpha_N]^T t + [\beta_1 \ldots \beta_N]^T \quad | \quad t \in \Re \right\}. \tag{3.2}$$

A key property of the timeline is that even though knowledge of $\mathcal{L}$ implies knowledge of the temporal alignment of the sequences, we can compute points on the timeline without knowing the sequences' alignment. This observation leads to a simple algorithm for reconstructing the timeline from dynamic features in the scene that are visible in two or more of the sequences.

Specifically, let $\mathbf{q}_1(f_1)$ be the instantaneous projection of a moving scene point in camera 1 at frame f_1, expressed in homogeneous 2D coordinates, as illustrated in Figure 3.1. Furthermore, let $\mathbf{q}_i(\cdot)$ be the trajectory traced by the point's projection in camera i and suppose that the fundamental matrix, $\mathcal{F}_{1i}$, between cameras 1 and i is known for all i, where $i = 2...N$. Observe that in this case we are considering the camera 1 as our reference camera. If the scene point is visible to all cameras when frame f_1 is captured by camera 1, we have the following constraint:

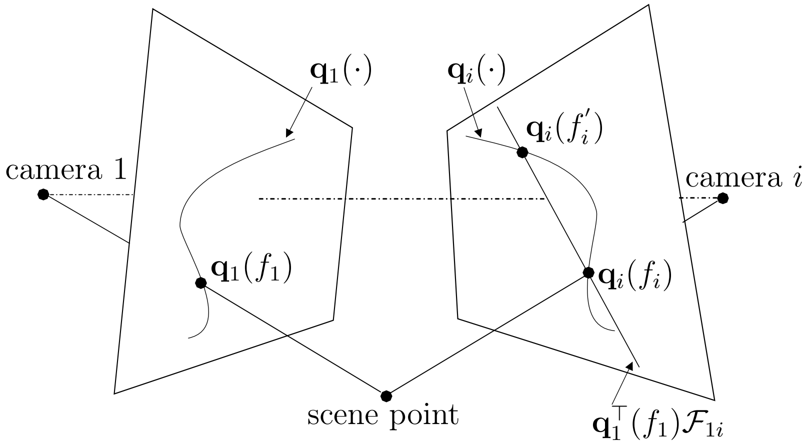


Figure 3.1: Geometry of the **Timeline Constraint**. In this two-camera example, the point's trajectory in camera i intersects the epipolar line, $\mathbf{q}_1^\top(f_1)\mathcal{F}_{1i}$, twice. Given the intersection points $\mathbf{q}_i(f_i)$ and $\mathbf{q}_i(f_i')$, we have the set $\mathcal{V}_{\mathbf{q}_1(f_1)} = \{ \ [f_1 \ \ f_i]^T, \ [f_1 \ \ f_i']^T \ \}$.

Timeline Constraint: The set

$$\mathcal{V}_{\mathbf{q}_1(f_1)} = \left\{ [f_1 \dots f_N]^T \ \mid \ \mathbf{q}_1^\top(f_1)\mathcal{F}_{1i}\mathbf{q}_i(f_i) = 0, \ i = 2 \dots N \right\}$$

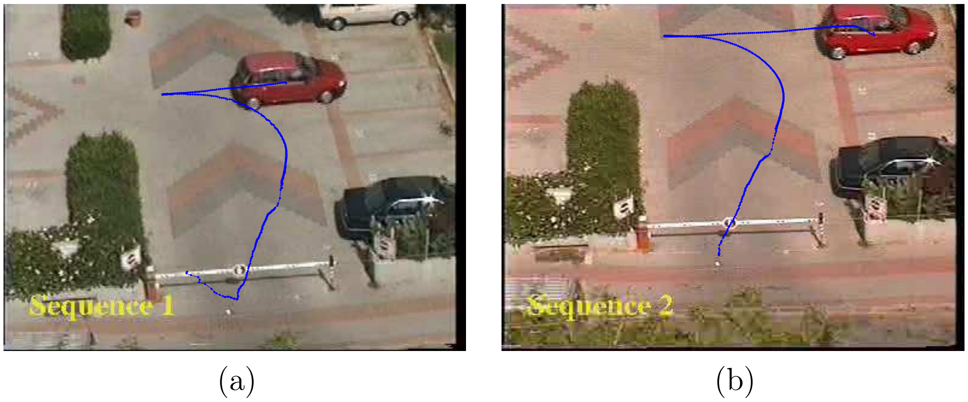
contains at least one point on the timeline $\mathcal{L}$.

Intuitively, the Timeline Constraint can be thought of as a procedure for generating a set $\mathcal{V}_{\mathbf{q}_1(f_1)}$ of "candidate" temporal alignments that is guaranteed to contain at least one point on the timeline. The constraint tells us that we can create such a set by (1) intersecting the epipolar line of $\mathbf{q}_1(f_1)$ in camera i with the trajectory $\mathbf{q}_i(\cdot)$, (2) recording the frame number(s) corresponding to each intersection point for camera i, and (3) generating temporal alignment vectors from the recorded frame numbers. To see why the Timeline Constraint holds, observe that if $[f_1 \dots f_N]^T$ is on the timeline it must represent the "true" temporal alignment between the frame f_1 of pixel $\mathbf{q}_1(f_1)$ and the remaining cameras. Hence, pixels $\mathbf{q}_1(f_1)$ and $\mathbf{q}_i(f_i)$ must satisfy the epipolar constraint equation, $\mathbf{q}_1^\top(f_1)\mathcal{F}_{1i}\mathbf{q}_i(f_i) = 0$. Since, by definition, the set $\mathcal{V}_{\mathbf{q}_1(f_1)}$ contains *all* temporal alignments that satisfy the epipolar constraint equation across the N cameras, it must also contain the true alignment, which is a point on the timeline $\mathcal{L}$. In this respect, the Timeline Constraint can be thought of as the converse of the epipolar constraint for the case of N unaligned sequences.

In order to apply the Timeline Constraint, we must know the fundamental matrices, $\mathcal{F}_{ij}$, describing the cameras' epipolar geometry between each pair (i, j) of cameras. In practice, we obtain an initial estimate of $\mathcal{F}_{ij}$ by finding "background features," i.e., points in the scene that remain stationary and are jointly visible by two or more cameras. Once the timeline $\mathcal{L}$ is reconstructed, that is, once the estimation of the temporal alignment is performed, we jointly optimize $\mathcal{L}$ and the parameters of the fundamental matrices that describe the scene geometry, by using a linear, iterative refinement procedure. We describe our temporal alignment algorithm in the next section, and consider in Chapter 4 the joint optimization of $\mathcal{L}$ and the pre-computed fundamental matrices $\mathcal{F}_{ij}$, which gives us the spatio-temporal alignment between the sequences.

3.2 The Temporal Alignment Algorithm

The Timeline Constraint leads directly to a voting-based algorithm for reconstructing the timeline of N sequences, which provides the temporal alignment. The algorithm operates in two phases. In the first phase, we choose one of the image sequences to be the reference sequence and use the instantaneous positions $\mathbf{q}_r(f_r)$ from each feature trajectory $\mathbf{q}_r(\cdot)$ of that



(a) (b)

Figure 3.2: (a) Trajectory of the car's pixel centroid in Sequence 1. (b) Trajectory of the car's pixel centroid in Sequence 2. The car was tracked by a simple blob tracker that relies on foreground-background detection to label all foreground pixels in each frame.

sequence together with the entire trajectories $\mathbf{q}_i(\cdot)$ of the other sequences to estimate $\mathcal{V}_{\mathbf{q}_r(f_r)}$ for each $\mathbf{q}_r(f_r)$. In the second phase, we fit an N-dimensional line $\mathcal{L}$ to the union of the estimated sets $\mathcal{V}_{\mathbf{q}_r(f_r)}$. Therefore, to fully specify this algorithm we must ask three questions:

1. How do we compute the feature trajectories $\mathbf{q}_i(\cdot)$, for $i = 1, ..., N$?

2. How do we estimate the set $\mathcal{V}_{\mathbf{q}_r(f_r)}$ for each $\mathbf{q}_r(f_r)$?

3. How do we compute the timeline $\mathcal{L}$?

To compute the feature trajectories $\mathbf{q}_i(\cdot)$, we use a two-frame feature tracker that is treated by our algorithm as a "black-box" responsible for returning a list of line segments of corresponding features for every pair of consecutive frames. Each line segment connects the location of a feature that was detected in some frame of the i-th sequence and was successfully tracked to the next frame, as illustrated in a real-world sequence in Figure 3.2. Importantly, our algorithm does not depend on a specific tracker. Thus, the choice of a particular tracking methodology depends exclusively on the scene's complexity and on the properties of the features of interest.

Next, to compute the set $\mathcal{V}_{\mathbf{q}_r(f_r)}$ for a given $\mathbf{q}_r(f_r)$, our algorithm uses the initial estimates of the fundamental matrices, $\mathcal{F}_{ij}$, between each pair (i, j) of cameras, as well as the line segments provided by the feature tracker. When a specific line segment intersects the epipolar line of $\mathbf{q}_r(f_r)$, it defines a possibly-fractional frame number, f_i, corresponding to the instant that $\mathbf{q}_r(f_r)$'s epipolar line intersects the image trajectory of a point in the scene. Hence, f_i is the i-th coordinate of a potential element of $\mathcal{V}_{\mathbf{q}_r(f_r)}$. To generate $\mathcal{V}_{\mathbf{q}_r(f_r)}$, we collect all the f_i coordinates computed for all sequences and concatenate them so that they form valid N-dimensional vectors, which represent candidate temporal alignments in a voting space. These steps are illustrated in Figure 3.4a-d for the two-camera example of Figure 3.2.

Note that according to the algorithm described above, our approach may result in a large number of intersections of the epipolar line of $\mathbf{q}_r(f_r)$ with the line segments of the trajectories in each one of the $N - 1$ cameras, since we may have several possible ways of "concatenating" the computed f_i coordinates into an N-dimensional vector. However, to avoid including an exponential number of vectors in $\mathcal{V}_{\mathbf{q}_r(f_r)}$, we only include vectors that are *consistent* with the cameras' epipolar geometry. In particular, let $[f_1 \ldots f_N]^T$ be a candidate vector for a set of N cameras, where f_1 represents the temporal coordinate of a feature position in the reference camera, that is, $\mathbf{q}_r(f_r) = \mathbf{q}_1(f_1)$. Given fundamental matrices with an average error of e pixels, where the error of a fundamental matrix is measured as the average of the distances between background feature projections in the image plane of the reference camera and their corresponding epipolar lines, we assume that the afore-mentioned candidate vector is *consistent* with the cameras' epipolar geometry only if

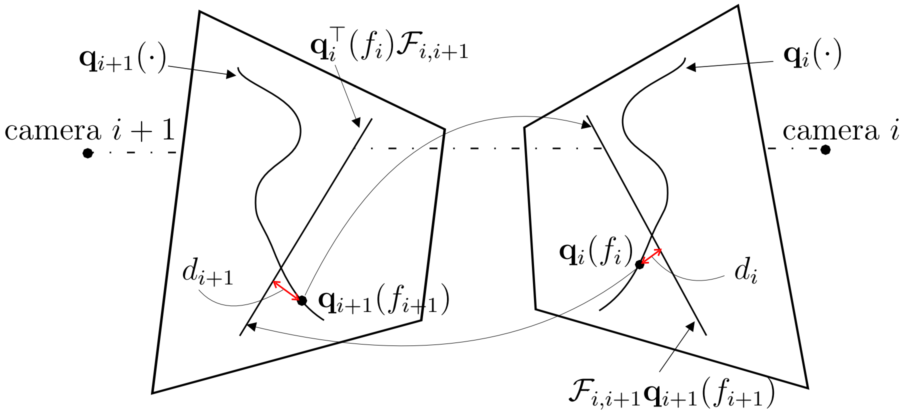

$\mathbf{q}_{i+1}(\cdot)$
$\mathbf{q}_i^\top(f_i)\mathcal{F}_{i,i+1}$
$\mathbf{q}_i(\cdot)$
camera $i+1$
camera i
d_{i+1}
$\mathbf{q}_{i+1}(f_{i+1})$
$\mathbf{q}_i(f_i)$
d_i
$\mathcal{F}_{i,i+1}\mathbf{q}_{i+1}(f_{i+1})$

Figure 3.3: Two intersection points $\mathbf{q}_i(f_i)$ and $\mathbf{q}_{i+1}(f_{i+1})$ of cameras i and $i+1$, respectively, are considered by our approach as potential representations of the same scene point only if $d_i \leq e$ and $d_{i+1} \leq e$, where d_i and d_{i+1} are distance values that measure how close $\mathbf{q}_i(f_i)$ and $\mathbf{q}_{i+1}(f_{i+1})$ are to each others' epipolar lines and e is the fundamental matrices' average error.

the intersection points that defined each pair of its consecutive temporal coordinates (f_i, f_{i+1}), for $2 \leq i \leq N-1$, satisfy: $d_i \leq e$ and $d_{i+1} \leq e$, where d_i and d_{i+1} are illustrated in Figure 3.3 and are distance values that measure how close the intersection points that defined f_i and f_{i+1} are to each others' epipolar lines.

Therefore, given a set of N cameras, our approach evaluates the constraints $d_i \leq e$ and $d_{i+1} \leq e$ for $N-2$ times. If all the $N-2$ evaluations performed obtain a positive answer, the candidate vector is considered as a potential inlier and is added to the voting space, otherwise it is rejected. Note that our procedure does not test those constraints for each possible pair of temporal coordinates, but rather it considers consecutive temporal coordinates in the candidate vector. That is, we make use of the transitive property that allows us to go from step to step in our reasoning process for inferring that a candidate point is consistent with the cameras' epipolar geometry.

For instance, consider that the intersection points that defined f_i and f_{i+1} are, according to our approach (see Figure 3.3), potential representations of a specific scene point $\mathbf{Q}$. If the intersection points that defined f_{i-1} and f_i potentially represent a scene point $\mathbf{P}$, we may infer that $\mathbf{P}$ and $\mathbf{Q}$ are probably the same point, since the intersection point of f_i is considered in both cases. Moreover, we may also conclude that the intersection points of f_{i-1} and f_{i+1} are probable representations of the same scene point. By using this reasoning process along consecutive temporal coordinates of the candidate point, we ensure that it is a potential representation of the temporal misalignment between the cameras. Note that our concatenation procedure is conservative, i.e., it guarantees that the set of vectors generated will be a superset of $\mathcal{V}_{\mathbf{q}_r(f_r)}$.

The set of candidate temporal alignments is the union of the sets $\mathcal{V}_{\mathbf{q}_r(f_r)}$ for all $\mathbf{q}_r(f_r)$. In general, this union will contain a large number of outliers, as illustrated in Figure 3.4d. To reconstruct the timeline in the presence of outliers, we use the RANSAC algorithm (Fischler and Bolles, 1981), which is described in detail in Appendix A.

RANSAC can be regarded as an algorithm for robust fitting of models in the presence of many data outliers (Fischler and Bolles, 1981). Since it gives us the opportunity to evaluate any estimate of a set of parameters no matter how accurate the method that generated this solution might be, the RANSAC method represents an interesting approach to the solution of many computer vision problems (Cantzler, 2004). The algorithm randomly chooses a pair of candidate temporal alignments to define the timeline $\mathcal{L}$, and then computes the total number of candidates that fall within an ϵ-distance of this line. These two steps are repeated for a number of iterations. Provided sufficient repetitions are performed, RANSAC is expected to identify solutions computed from outlier-free data. Therefore, the two critical parameters of

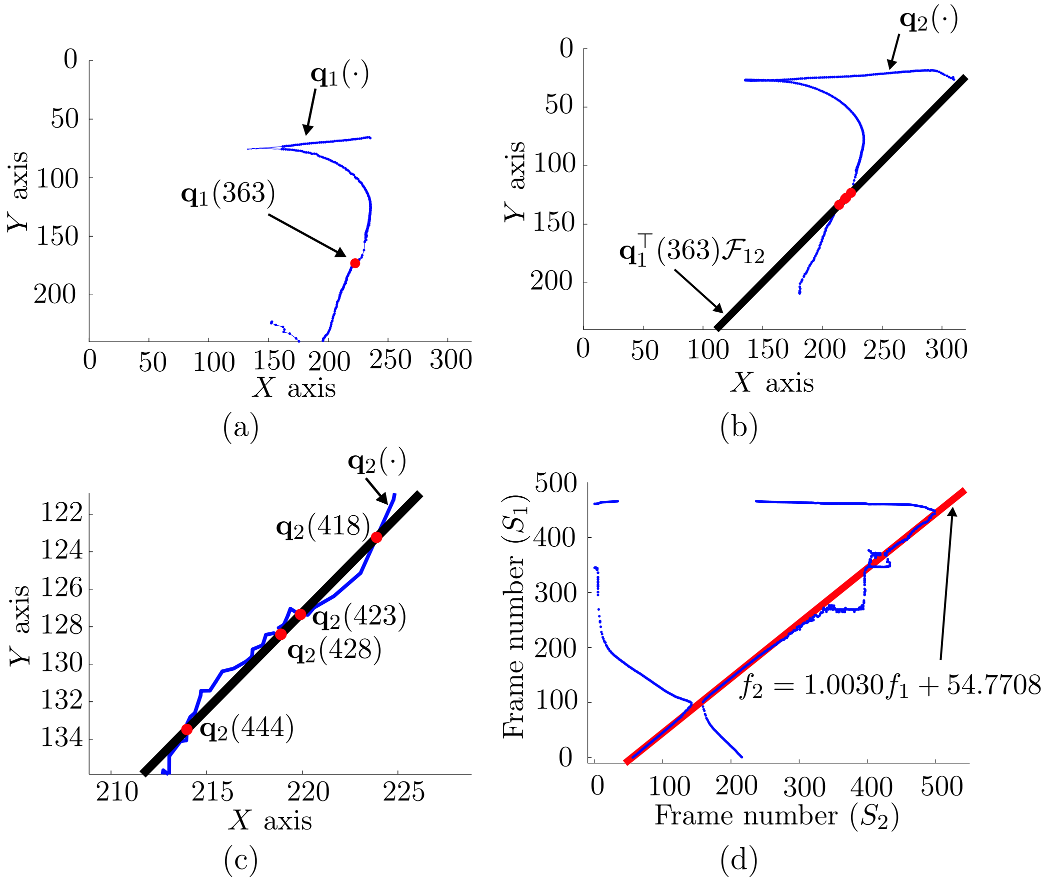


Figure 3.4: (a) Trajectory of a feature in Sequence 1 of the *Car* dataset, which is presented in Chapter 5 and illustrated in Figure 3.2. The feature was the centroid of all pixels labeled as "foreground" by a color-based foreground-background detector. (b) Trajectory of the foreground pixel centroid in Sequence 2 of the dataset. Also shown is the epipolar line corresponding to pixel $\mathbf{q}_1(363)$ in (a). (c) Magnified view of the trajectory/epipolar line intersection in (b). The individual line segments connecting feature locations in adjacent frames are now visible. Note that the epipolar line of $\mathbf{q}_1(363)$ intersects multiple line segments along the trajectory. (d) Exploiting the Timeline Constraint for two-sequence alignment. Each point represents a candidate temporal alignment, i.e., an element of $\mathcal{V}_{\mathbf{q}_1(f_1)}$ for the feature location, $\mathbf{q}_1(f_1)$, in (a). The reconstructed timeline, drawn as a solid line, describes the temporal alignment of the two sequences in the *Car* dataset.

the algorithm are the number z of RANSAC iterations and the distance ϵ. To determine z, we use the formula

$$z = \left\lceil \frac{\log(1-p)}{\log(1-r^2)} \right\rceil, \tag{3.3}$$

where p is the probability that at least one of our random selections is an error-free set of candidates and r is the probability that a randomly-selected candidate is an inlier.

Equation (3.3) expresses the fact that z should be large enough to ensure that, with probability p, at least one randomly-selected pair of candidates is an inlier. We used $p = 0.99$ and $r = 0.05$ ($z = 1840$ iterations) for all experiments, which are conservative values that lead to accurate results in our experiments, as demonstrated in Chapter 5. To compute the distance ϵ, we observe that ϵ can be thought of as a bound on the distance between detected feature locations in the input cameras and their associated epipolar lines. This allows us to approximate ϵ by the average distance between static features in the scene and their associated epipolar lines. Next, we summarize our algorithm for recovering the temporal alignment between multiple video sequences.

Algorithm 1 The Temporal Alignment Algorithm

```
▷ Input
$N$; {Number of cameras.}
$\mathcal{F}_{ij}$; {Fundamental matrices between cameras $i$ and $j$.}
▷ Output
$[\alpha_1 ... \alpha_N]^\top$; {Temporal dilation parameters of the timeline $\mathcal{L}$.}
$[\beta_1 ... \beta_N]^\top$; {Temporal shift parameters of the timeline $\mathcal{L}$.}

BEGIN

{Step 1 - Generate the voting space: $\bigcup \mathcal{V}_{\mathbf{q}_r(f_r)}$, $\forall \mathbf{q}_r(f_r)$.}
for $i \leftarrow 1$ to $N$ do
  Compute feature trajectories $\mathbf{q}_i(\cdot)$.
end for
{Consider the camera 1 as the reference camera.}
for (each instantaneous position $\mathbf{q}_1(f_1)$) do
  for $i \leftarrow 2$ to $N$ do
    for (each line segment in sequence $i$) do
      if (the epipolar line $\mathbf{q}_1^\top(f_1)\mathcal{F}_{1i}$ intersects the line segment) then
        Obtain the frame number $f_i$ of the intersection's point.
        if (the corresponding pixels of $(f_1,f_i)$ are consistent
        with the cameras' epipolar geometry) then
          Store $f_i$ in the intersections' vector of sequence $i$.
        end if
      end if
    end for
  end for
  Construct the set $\mathcal{V}_{\mathbf{q}_1(f_1)}$ by collecting all the $f_i$ coordinates
  computed and concatenating them so that they form
  valid $N$-dimensional vectors $[f_1 ... f_N]$.
end for
Generate the voting space by performing the union of the sets $\mathcal{V}_{\mathbf{q}_1(f_1)}$.
{Step 2 - Fit the timeline $\mathcal{L}$ to the union of the estimated sets $\mathcal{V}_{\mathbf{q}_1(f_1)}$.}
Apply the RANSAC algorithm to the voting space in order to
determine the data set that best fits the searched timeline $\mathcal{L}$.
Apply the least-squares method over the data set estimated by
RANSAC to compute the timeline parameters: $[\alpha_1 ... \alpha_N]^\top$ and $[\beta_1 ... \beta_N]^\top$.

return($[\alpha_1 ... \alpha_N]^\top$,$[\beta_1 ... \beta_N]^\top$).

END
```

Chapter 4

Spatio-Temporal Alignment

> Time and space are modes by which we think and not conditions in which we live.
>
> *Albert Einstein*

While images of a dynamic scene may contain stationary points in the background, these points cannot be expected to represent the majority of detected features. Any procedure that attempts to estimate the geometric relations between the views from those features alone is likely to ignore a significant portion of the available image information. In practice, this will cause errors in the computed fundamental matrices that encapsulate the geometric relations and, ultimately, in the reconstructed timeline. In this chapter we show how to refine the pre-computed fundamental matrices $\mathcal{F}_{ij}$ and the timeline $\mathcal{L}$ by incorporating all features detected in the sequences, i.e., both the tracked dynamic features and the static features detected on the background. Without loss of generality, we assume that the camera represented by the number 1 is our reference camera.

4.1 The Refinement Algorithm

Even though the solution presented in Chapter 3 for temporally aligning multiple video sequences is robust, it is not very accurate: it does not use any information from the dynamic features to refine the *spatial* alignment between sequences and even the *temporal* alignment is affected by errors in the initial spatial alignment, because the RANSAC threshold for regarding a point in the voting space as an inlier is set proportional to the magnitude of those errors.

In the following, we present a method that can refine both the temporal and the spatial transformations between sequences, using information from all features available. Firstly, we present its derivation for the case of scenes monitored by two distinct viewpoints ($N = 2$) and, next, we present the general ideas behind its application for $N > 2$.

4.1.1 Two-View Refinement

The geometric basis of the refinement method in the case of two different viewpoints is presented in Figure 4.1. The multilinear tensor that encapsulates the geometric relations between the two views is represented by a fundamental matrix. To make the problem linear, we approximate each tracked trajectory $\mathbf{q}_2(\cdot)$ with a polygonal spline $\mathbf{s}_2(\cdot)$ (represented by the red line segments in Figure 4.1) using a method proposed by Horst and Beichl that guarantees an upper bound on the difference between the length of the original curve and the length of its polygonal approximation (Horst and Beichl, 1997). Importantly, we parameterize these polygonal approximations in a way that is consistent with the (temporal) parameterization of the original curves, *i.e.*, each point that lies both on the original curve and on the ap-

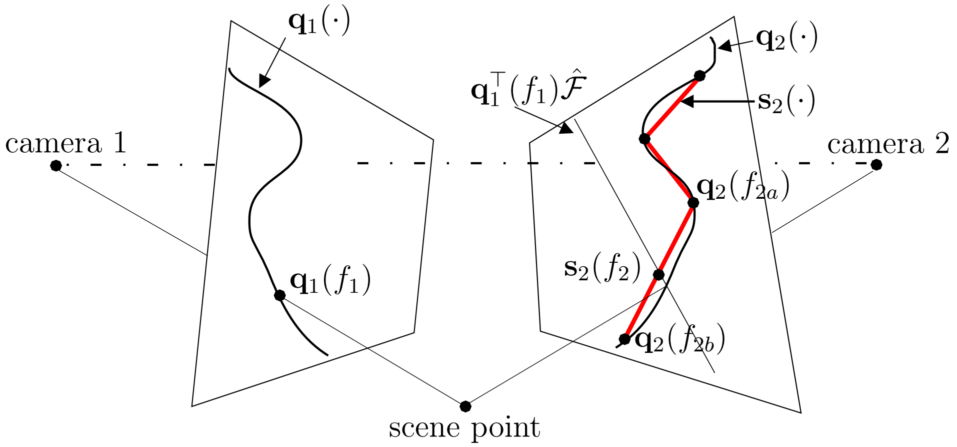


Figure 4.1: Geometry of our refinement method.

proximation keeps its original coordinate, so that each linear spline segment has endpoints with well–defined temporal coordinates f_{2a} and f_{2b}.

The key steps of our method to refine the current estimates for the spatial parameters represented by the entries of the fundamental matrix $\hat{\mathcal{F}}$, and the temporal parameters $\hat{\alpha}$ and $\hat{\beta}$ of the timeline, are:

1. Create the epipolar lines $\mathbf{q}_1^\top(f_1)\hat{\mathcal{F}}$ in camera 2 from their corresponding instantaneous feature positions $\mathbf{q}_1(f_1)$ in camera 1, by using $\hat{\mathcal{F}}$.

2. Intersect the epipolar lines $\mathbf{q}_1^\top(f_1)\hat{\mathcal{F}}$ with the polygonal trajectory approximations $\mathbf{s}_2(t_2)$ in camera 2.

3. Screen the resulting intersections $\mathbf{s}_2(f_2)$ for *consistency* with the current estimates for $\hat{\alpha}$ and $\hat{\beta}$.

4. Use only the *consistent* intersections to generate algebraic equations that jointly constrain the (unknown) transformation parameters $\mathcal{F}$, α and β.

More specifically, according to Equation (3.1), an estimated temporal transformation with parameters $\hat{\alpha}$ and $\hat{\beta}$ implies that any instantaneous feature $\mathbf{q}_1(f_1)$ in camera 1 should correspond to a feature in camera 2 whose temporal coordinate is

$$f_2 = \hat{\alpha} f_1 + \hat{\beta}. \tag{4.1}$$

Thus, an intersection $\mathbf{s}_2(f_2)$ in camera 2 (see Figure 4.1) between an epipolar line $\mathbf{q}_1^\top(f_1)\hat{\mathcal{F}}$ and an approximated trajectory segment whose endpoints have temporal coordinates f_{2a} and f_{2b} is *consistent* with the current estimated temporal alignment only if

$$f_{2a} < \hat{\alpha} f_1 + \hat{\beta} < f_{2b}. \tag{4.2}$$

Every intersection that satisfies the consistency condition in Eq. (4.2) yields a linear constraint on α, β, and the entries of $\mathcal{F}$. To obtain this constraint, we note that an arbitrary point $\mathbf{s}_2(f_2)$ on an intersected spline segment $\overline{\mathbf{q}_2(f_{2a})\mathbf{q}_2(f_{2b})}$ (Figure 4.1) is, according to the segment's parameterization, given by

$$\mathbf{s}_2(f_2) = \mathbf{q}_2(f_{2a}) + (f_2 - f_{2a})\frac{\mathbf{q}_2(f_{2b}) - \mathbf{q}_2(f_{2a})}{(f_{2b} - f_{2a})}. \tag{4.3}$$

Observe that if $f_2 = f_{2a}$ then $\mathbf{s}_2(f_2) = \mathbf{q}_2(f_{2a})$, which means that the epipolar line $\mathbf{q}_1^\top(f_1)\hat{\mathcal{F}}$ intersects the endpoint $\mathbf{q}_2(f_{2a})$. Similarly, if $f_2 = f_{2b}$, then $\mathbf{s}_2(f_2) = \mathbf{q}_2(f_{2b})$, meaning that the epipolar line $\mathbf{q}_1^\top(f_1)\hat{\mathcal{F}}$ intersects the other endpoint $\mathbf{q}_2(f_{2b})$.

Now, consider that $\mathbf{s}_2(f_2)$ in camera 2 represents the corresponding point of $\mathbf{q}_1(f_1)$ in camera 1. Given that assumption, those points together must satisfy the following equation

$$\mathbf{q}_1^\top(f_1)\mathcal{F}\mathbf{s}_2(f_2) = 0. \tag{4.4}$$

Observe that we have considered $\mathcal{F}$ instead of $\hat{\mathcal{F}}$ in Equation (4.4), where $\mathcal{F} = \hat{\mathcal{F}} + \Delta\mathcal{F}$, *i.e.*, the entries of $\mathcal{F}$ represent the spatial parameters obtained after the refinement process, $\hat{\mathcal{F}}$ is the current estimate of the fundamental matrix and $\Delta\mathcal{F}$ is the refinement term computed by our method. Substituting Equation (4.3) into Equation (4.4), we obtain:

$$\mathbf{q}_1^\top(f_1)\mathcal{F}\left\{\mathbf{q}_2(f_{2a}) + (f_2 - f_{2a})\frac{\mathbf{q}_2(f_{2b}) - \mathbf{q}_2(f_{2a})}{(f_{2b} - f_{2a})}\right\} = 0. \tag{4.5}$$

Equation (4.5) may be rewritten in a more concise manner as follows:

$$\mathbf{q}_1^\top(f_1)\left\{\mathcal{F}\mathbf{k}f_2 + \mathcal{F}\mathbf{m}\right\} = 0, \tag{4.6}$$

where

$$\begin{aligned} \mathbf{k} &= \frac{\mathbf{q}_2(f_{2b}) - \mathbf{q}_2(f_{2a})}{(f_{2b} - f_{2a})}. & (4.7) \\ \mathbf{m} &= \mathbf{q}_2(f_{2a}) - f_{2a}\mathbf{k}. & (4.8) \end{aligned}$$

Now, by considering $f_2 = \alpha f_1 + \beta$, where $\alpha = \hat{\alpha} + \Delta\alpha$ and $\beta = \hat{\beta} + \Delta\beta$, *i.e.*, α and β are the temporal parameters obtained after the refinement process, $\hat{\alpha}$ and $\hat{\beta}$ are the current estimates and $\Delta\alpha$, $\Delta\beta$ are the refinement terms, and similarly by writing $\mathcal{F} = \hat{\mathcal{F}} + \Delta\mathcal{F}$, the factor enclosed by curly brackets in the Equation (4.6) becomes

$$\begin{aligned}\hat{\mathcal{F}}(f_1\hat{\alpha}\mathbf{k}+\hat{\beta}\mathbf{k}+\mathbf{m}) &+ f_1\hat{\mathcal{F}}\mathbf{k}\Delta\alpha+\hat{\mathcal{F}}\mathbf{k}\Delta\beta+\\ &+ \Delta\mathcal{F}(f_1\hat{\alpha}\mathbf{k}+\hat{\beta}\mathbf{k}+\mathbf{m})+\\ &+ f_1\Delta\alpha\Delta\mathcal{F}\mathbf{k}+\Delta\beta\Delta\mathcal{F}\mathbf{k}.\end{aligned}$$

Disregarding the second-order terms $f_1\Delta\alpha\Delta\mathcal{F}\mathbf{k}$ and $\Delta\beta\Delta\mathcal{F}\mathbf{k}$, we obtain the following linear constraint on $\Delta\mathcal{F}$, $\Delta\alpha$ and $\Delta\beta$:

$$\mathbf{q}_1^\top(f_1)\left\{f_1\hat{\mathcal{F}}\mathbf{k}\Delta\alpha+\hat{\mathcal{F}}\mathbf{k}\Delta\beta+\Delta\mathcal{F}\mathbf{h}\right\}=-\mathbf{q}_1^\top(f_1)\hat{\mathcal{F}}\mathbf{h}, \qquad \text{where} \quad \mathbf{h}=(f_1\hat{\alpha}+\hat{\beta})\mathbf{k}+\mathbf{m}. \tag{4.9}$$

After straightforward algebraic manipulation, Equation (4.9) may be rewritten as the (inner) product of two vectors: a 11-element row vector that contains only known coefficients and a 11-element column vector that contains the 9 unknown coefficients of $\Delta\mathcal{F}$ followed by the scalar unknowns $\Delta\alpha$ and $\Delta\beta$. Linear constraints of this form yielded by all the *temporally consistent* intersections between epipolar lines and approximated trajectories are finally assembled into an over-constrained linear system. Importantly, traditional linear constraints of the type used in the eight-point algorithm can also be added to this system, just by concatenating two zeros at the end of their coefficient vectors, as the coefficients of the temporal parameters $\Delta\alpha$ and $\Delta\beta$. Therefore, our solution allows us to use all avaliable constraints both from static features and from dynamic features in order to refine the estimated spatio-temporal alignment.

The set of equations that we construct are of the form $\mathcal{A}_{n\times 11}\mathbf{x}_{11\times 1}=\mathbf{b}_{n\times 1}$, where the number of linear constraints n is frequently much larger than the

11 unknowns. Our task now consists in finding the best solution $\mathbf{x}$ for that linear system. There are many different techniques for solving such a system (Press et al., 1988; Atkinson, 1989; Tomasi, 2000; Hefferon, 2001) and a good way to find its solution is by using the Singular Value Decompostion (SVD) (Press et al., 1988), although the linear least-squares methods represent also very interesting alternatives (Atkinson, 1989). By using one of the abovementioned numerical methods, our spatio-temporal approach computes the system's solution in an iterative way, until the convergence to zero of the unknowns $\Delta\mathcal{F}$, $\Delta\alpha$ and $\Delta\beta$.

4.1.2 N-View Refinement

Consider now a dynamic scene viewed by N distinct cameras, where $N > 2$, and suppose that our reference camera is the camera labeled by the number 1, which is denoted by c_1. In this case, we may simply use the two-view refinement technique previously described for each pair of cameras (c_1,c_i), where c_i is the i-th camera, for $i = 2, ..., N$. Thus, by combining the computed equations $t_i = \alpha_i t_1 + \beta_i$ with refined parameters α_i and β_i, we may obtain new equations that capture the temporal relation between any two arbitrary sequences i and j.

As far as our methodology for refining the statio-temporal alignment between N different video sequences is concerned, one may argue that the use of other tools such as the trifocal and quadrifocal tensors could be also considered, since they represent natural extensions of the fundamental matrix in the case of three and four views, respectively. Because of the added stability of a third or even a fourth view, and the fact that they constrain the position of reconstructed points in space more tightly, use of the trifocal and quadrifocal tensors should lead to greater accuracy than two-view techniques (Hartley, 1998). This hypothesis is supported by the results of Heyden (Heyden, 1995a,b). In order to evaluate the use of those alternative tools, we have derived and tested a three-view refinement methodology similarly to the previous one derived for the two-view case. The main difference of this alternative approach relates to the multilinear tensor (a trifocal tensor instead of a fundamental matrix) that encapsulates the geometric relations between the views. Our experiments showed that this strategy is not effective, since that the optimization of the temporal and spatial parameters is quite unstable. One of the main impediments to use the trifocal tensor (and the quadrifocal tensor) is its overparametrization (Hartley, 1998), using 29 components of the tensor to describe a geometric configuration that depends only on 18 parameters. This is probably the main reason for the inaccuracies of our results and the observed instability in the optimization process.

Chapter 5

Experimental Results

> No amount of experimentation can ever prove me right; a single experiment can prove me wrong.
>
> *Albert Einstein*

In this chapter, we present and discuss several experimental results with real-world and synthetic sequences. Firstly, in Section 5.1, we illustrate the applicability of our approach by testing it on several challenging two- and three-view datasets of real-world dynamic scenes. Scenarios that may be critical for most of the current spatio-temporal alignment methodologies are successfully handled by our approach, such as, situations where a reliable feature tracking cannot be performed over the entire sequence, the initial estimates of the cameras' epipolar geometry is inaccurate, the video sequences have large temporal misalignments and the scene points move along three-dimensional, overlapping and cyclical trajectories.

However, although these experiments with real-world sequences demonstrate how effective our method can be in some common and challenging scenarios, it is not possible to perform from their results a careful analysis of the scalability, efficiency and accuracy of our approach, since that many critical variables, such as the error in the initial estimate of the cameras' epipolar geometry, had their values tested in a limited range.

Therefore, to obtain additional experimental results for a better evaluation of our method, we have performed experiments with synthetic sequences of an artificial scene, presented in Section 5.2, where we could simulate and control some of the main parameters that affect the results of our approach. In particular, based on that simulation, we obtained quantitative measurements of the quality of the estimated spatio-temporal alignments as functions of three key factors: (1) the accuracy of the initial estimates of the fundamental matrices that capture the geometric relations between the views (2) the amount of noise in the tracking system and (3) the number of tracked features that are considered by the method.

5.1 Real-world Sequences

To demonstrate the applicability of our timeline reconstruction algorithm, we tested it on various challenging two- and three-view real-world datasets. Image dimensions in all datasets were about 320×240 pixels. The sequences represented a wide variety of conditions, including sequences that ranged from 55 to 605 frames; temporal misalignments of 21 to 285 frames; relative frame rates between 1 and 2; image quality that ranged from quite high (i.e., sequences captured by laboratory-based color cameras) to rather low (i.e., clips from a low-quality, MPEG-compressed video of a broadcast TV signal); and object motions ranging from several pixels per frame to less than a pixel.

Since no single tracker was able to handle all of our datasets, and since our algorithm does not depend on a specific tracker, we experimented with several—a simple color-based blob tracker, a blob tracker based on background subtraction, and the WSL tracker (Jepson et al., 2003). In each case, we treated the tracker as a “black box” that returned a list of corresponding features for every pair of consecutive frames.

Alignment accuracy was evaluated by measuring the average temporal misalignment. This is the average difference between the computed time of each frame and the frame’s “ground-truth” time, i.e., when it was actually captured. Since our sequences were acquired with unsynchronized cameras, the ground-truth time of each frame could only be known to within ± 0.5 frames. This is because even if we could perfectly align the sequences at frame resolution, corresponding frames could have been captured up to 0.5 frame intervals apart. This lower bound on ground-truth accuracy is critical in evaluating the presented results.

5.1.1 Two-view *Car* Dataset

As a first test, we applied our technique to a two-view sequence used by Caspi and Irani (Caspi and Irani, 2000) for evaluating their method (Figure 5.1). The data was acquired by two cameras with identical frame rate of 25fps, implying a unit ground-truth temporal dilation ($\alpha = 1$). The ground-truth temporal shift was $\beta = 55 \pm 0.5$ frames.

Most frames in the resulting sequences contain a single rigid object (a car) moving over a static background (a parking lot), along a fairly smooth trajectory. We therefore used a simple blob tracker that relied on foreground-background detection to label all foreground pixels in each frame. The centroid of the foreground pixels was the only “feature” detected and tracked (Figures 3.2(a) and 3.2(b)). To compute the cameras’ fundamental matrix we used the normalized eight-point algorithm (Hartley, 1997a), which was provided with twenty six correspondences between background pixels in the two views illustrated in Figure 5.2.

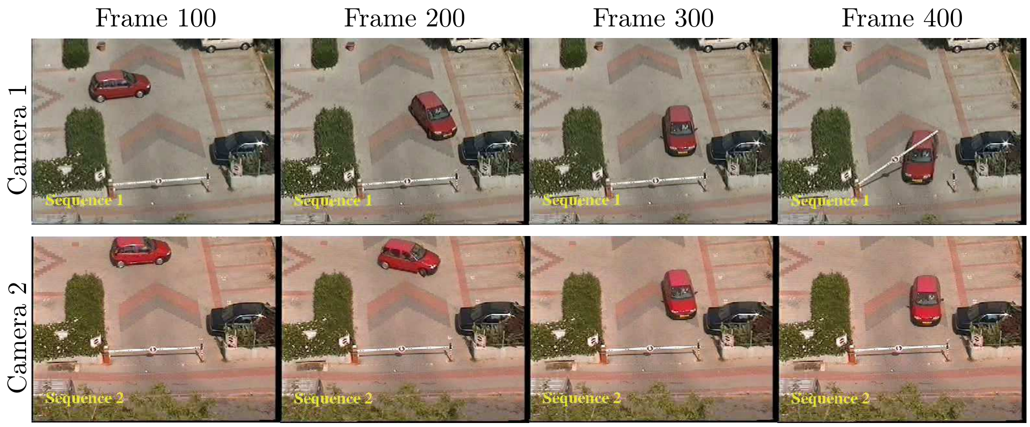


Figure 5.1: Four representative frames (100, 200, 300, 400) from the cameras 1 and 2, of the two-view *Car* dataset (Caspi and Irani, 2000). We can identify the spatial misalignment by observing near image boundaries, where different static objects are visible in each sequence. The temporal misalignment is easily identified by comparing the position of the gate in frames 400. This dataset, along with more experimental results and videos, are available at *http://www.dcc.ufmg.br/˜cardeal/research/timeline/* .

Figure 3.4(d) shows the timeline reconstructed using the RANSAC-based algorithm of Chapter 3, with the RANSAC parameter ϵ set to 2.0. The reconstructed timeline gives an average temporal misalignment of 0.66 frames, almost within the 0.5-frame uncertainty of the ground-truth measurements. By applying the refinement procedure of Chapter 4 we obtained updated values of $\alpha = 1.0027$ and $\beta = 54.16$ for the timeline coefficients. These coefficients correspond to an improved average temporal misalignment of 0.35 frames, i.e., below the accuracy of the ground-truth alignment. Note that these results are at least as accurate as those of Caspi and Irani, even though we are solving a less constrained problem (i.e., α is unknown and scene planarity is not required). Moreover, the results were obtained from raw results of a tracker that was not particularly robust (e.g., the centroid of the foreground pixels drifts off the moving car for approximately 30 frames in each sequence).

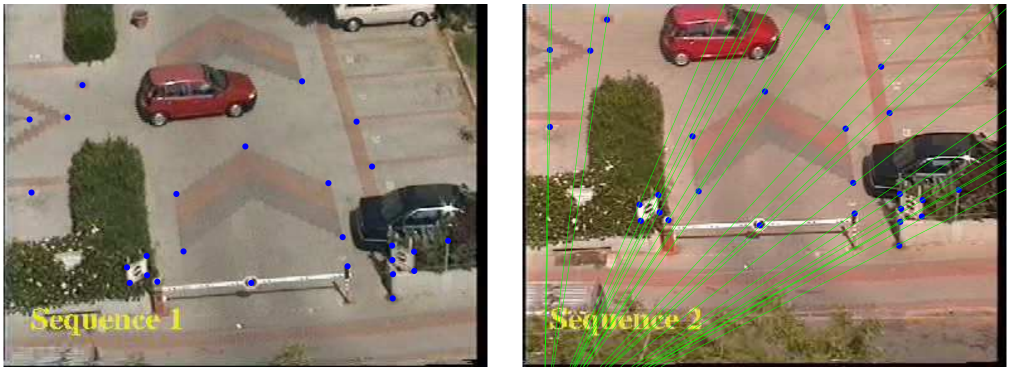


Figure 5.2: The recovered epipolar geometry for the two-view *Car* dataset. Points and their epipolar lines are displayed in each image for verification. Accuracy of the computed fundamental matrix can be appreciated by the closeness of each point to the epipolar line of its corresponding point.

5.1.2 Two-view ***Robots*** Dataset

In a second experiment, we used two cameras operating at 30fps to acquire images of four small robots, as they executed small random movements on two planes (Figure 5.3). The ground-truth timeline coefficients were $\alpha = 1$ and $\beta = -284.5 \pm 2$. We used a uniform-color blob tracker to track these robots between consecutive frames. The resulting data was challenging for four reasons. First, the robots' inter-frame motion was imperceptibly small (roughly 0.25 pixels per frame), making precise manual alignment by a human observer virtually impossible. Second, the temporal shift of the sequences was large, making it inefficient to find this shift via exhaustive search. Third, the uniformly-colored regions on each robot were small, causing our tracker to generate fragmented and noisy trajectories. Fourth, the robot's motion was designed to produce trajectories that self-intersect and that are non-smooth, complicating the shape of each blob's trajectory.

The timeline reconstructed with $\epsilon = 2.0$ prior to refinement is shown in Figure 5.4. This line gives an average temporal misalignment error of

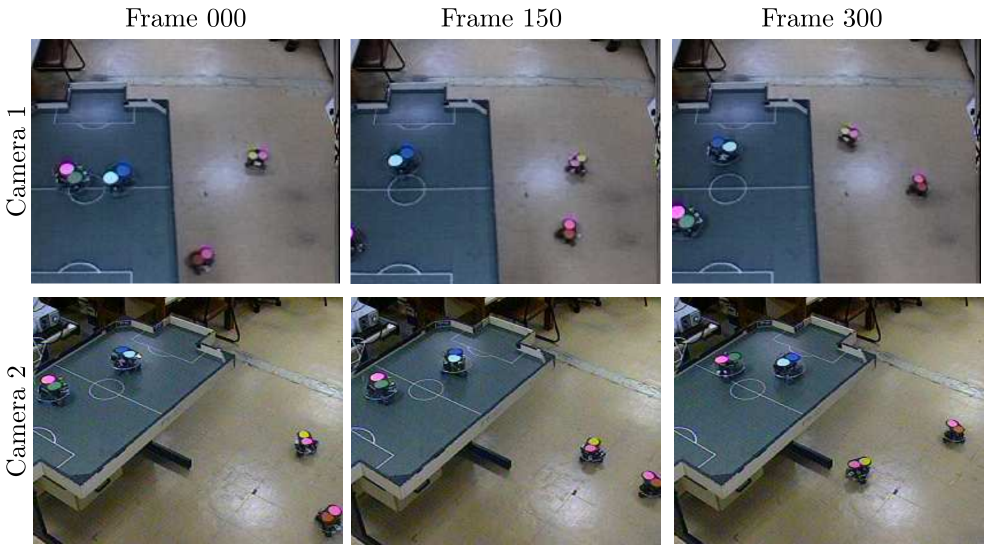


Figure 5.3: Three representative frames (000, 150, 300) from the cameras 1 and 2, of the two-view *Robots* dataset. The spatial misalignment can be easily identified by observing the distinct orientations of the robots' soccer field. On the other hand, the temporal misalignment can be noticeable by comparing the position of the robot with the dark green circle in frames 300. This dataset, along with more experimental results and videos, are available at *http://www.dcc.ufmg.br/˜cardeal/research/timeline/* .

5.84 frames. Our refinement stage reduced this error to 4.43 frames, with $\alpha = 1.015$ and $\beta = -286.89$. Given the robots' image velocity, this translates to a misalignment of about one pixel. Figure 5.7 confirms that the computed alignment is quite good, despite the robots' slow motion and the tracker's poor performance.

5.1.3 Two-view *Juggling* Dataset

In this dataset, two people are observed by a wide-baseline camera pair while juggling five uniformly-colored balls (see Figure 5.5). Both sequences were acquired at a rate of 30fps. This dataset represents a difficult case for existing direct- or feature-based methods because (1) the trajectories of dif-

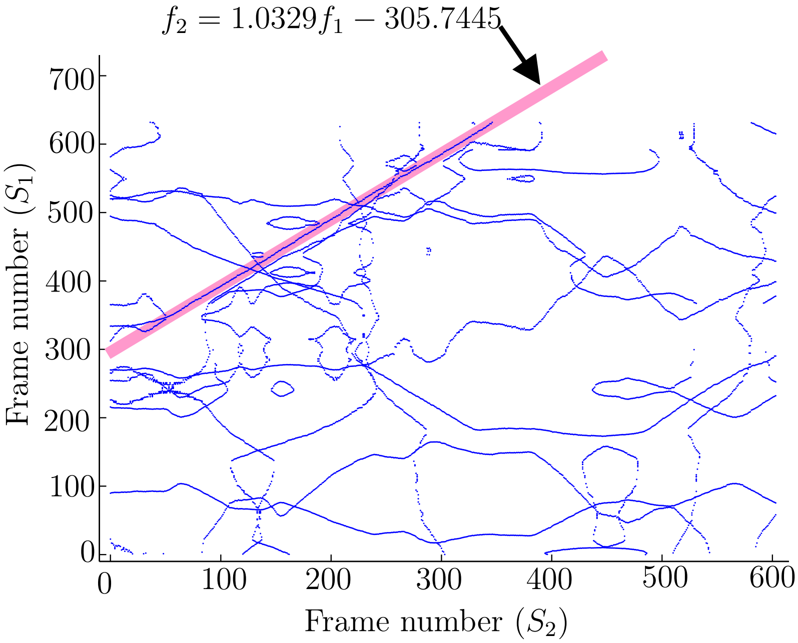


Figure 5.4: Voting space, timeline, and timeline equation recovered prior to refinement for the two-view *Robots* dataset. Each point is an element of $\mathcal{V}_{\mathbf{q}_1(f_1)}$ for some feature $\mathbf{q}_1(f_1)$ in sequence 1.

ferent balls nearly overlap in 3D, (2) individual trajectories are approximately cyclical, (3) image velocities are quite large, up to 9 pixels per frame, making long-range feature tracking difficult, and (4) the ground-truth temporal shift between the sequences is $\beta = -41 \pm 0.5$ frames, or about 1.5 periods of a ball's motion. This shift is likely to cause difficulties for techniques based on non-exhaustive search (Rao et al., 2003) or non-linear optimization (Caspi et al., 2002) because of the possibility of getting trapped in deep local minima.

To make the alignment problem even more challenging, we modified this dataset by deleting or adding frames to one of the sequences. These modifications were intended to simulate sequences with more than one frame rate (e.g., containing a slow-motion segment) and sequences that contain spuri-

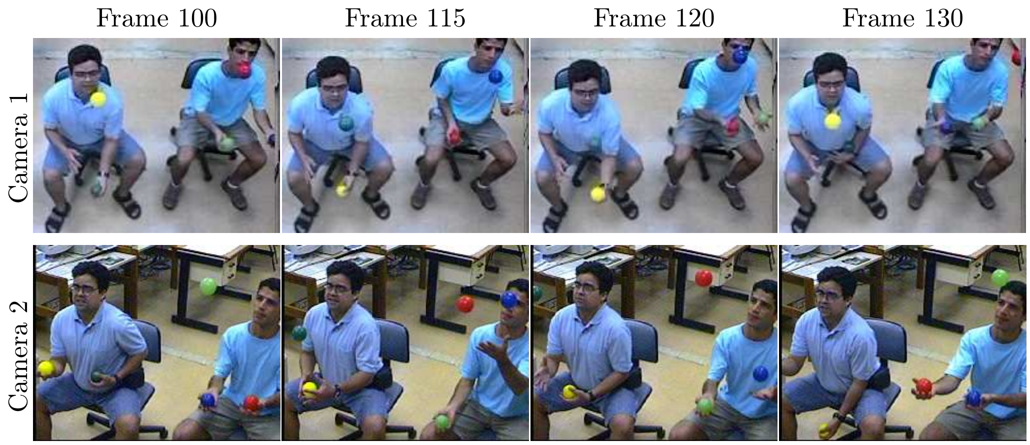
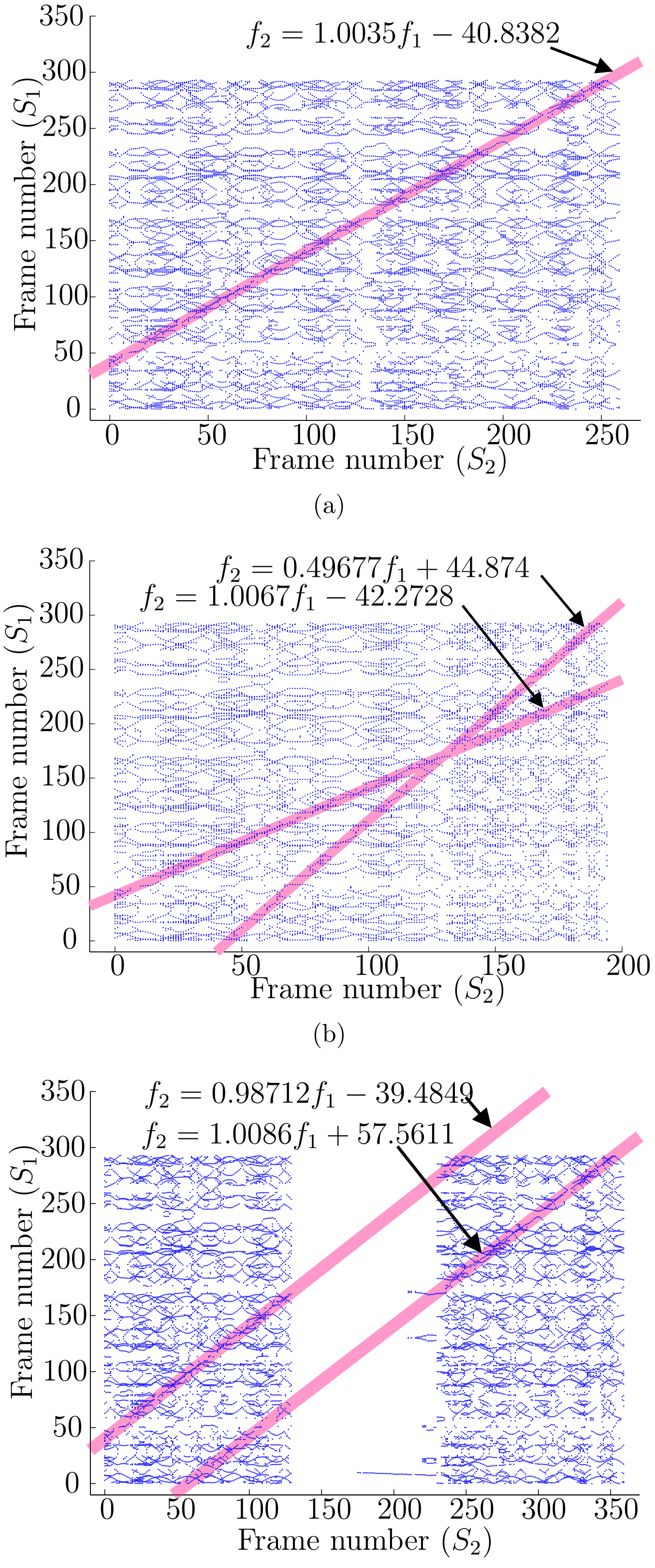


Figure 5.5: Four representative frames (100, 115, 120, 130) from the cameras 1 and 2, of the two-view *Juggling* dataset. This dataset, along with more experimental results and videos, are available at *http://www.dcc.ufmg.br/˜cardeal/research/timeline/* .

ous clips (e.g., a TV commercial). We used a uniform-color blob tracker to track four of the balls in each sequence, providing us with the location of four features per frame. No information about feature correspondences between cameras was given to the algorithm (i.e., color information was not used). Figures 5.6(a)-(c) show the reconstructed timelines before the refinement stage, with $\epsilon = 0.5$. The average temporal misalignment error was 0.75 frames for the original dataset. The refinement stage brought this error down to 0.26 frames, with $\alpha = 1.0004$ and $\beta = -40.80$.

Figure 5.7 confirms that the computed alignment was effectively retrieved and Figure 5.8 illustrates the distribution of distances of inlier votes from the reconstructed timeline for the *Car*, *Robots* and *Juggling* datasets, before and after the timeline refinement stage.



(a)

(b)

(c)

Figure 5.6: Voting spaces, timelines, and timeline equations recovered prior to refinement for the two-view *Juggling* dataset. (a) *Juggling* dataset without modification, (b) simulation of a sequence with more than one frame rate and (c) simulation of a sequence with spurious clips.

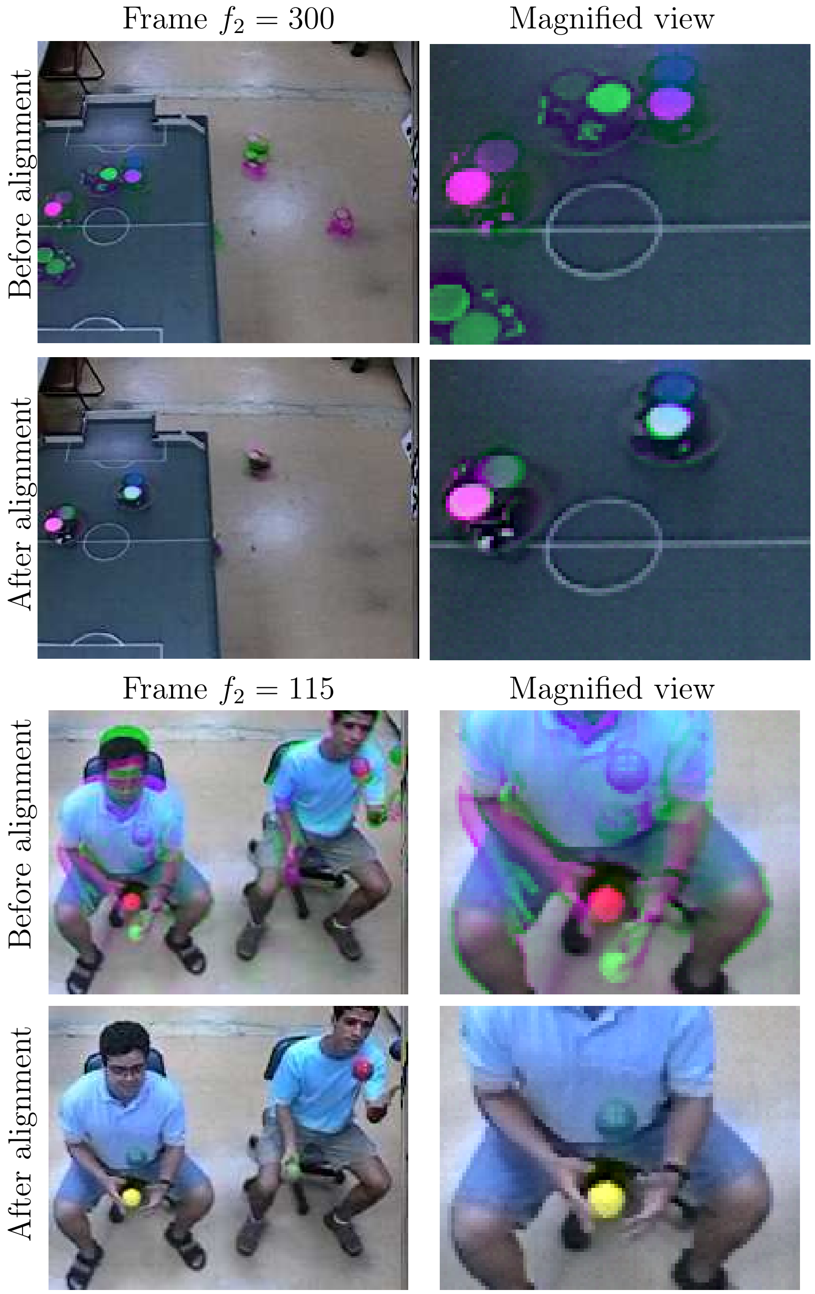


Figure 5.7: **Before alignment images** were created by superimposing the green band of a frame f_2 with the red and blue bands of frame $f_1^* = (f_2 - \beta^*)/\alpha^*$ using ground truth timeline coefficients α^* and β^*. **After alignment images** were created by replacing the green band of the images above them with that of frame $f_1 = (f_2 - \beta)/\alpha$, with α, β computed by our algorithm. For both types of images, deviations from the ground-truth alignment cause "double exposure" artifacts (i.e., when $f_1^* \neq f_2$ or $f_1^* \neq f_1$, respectively).

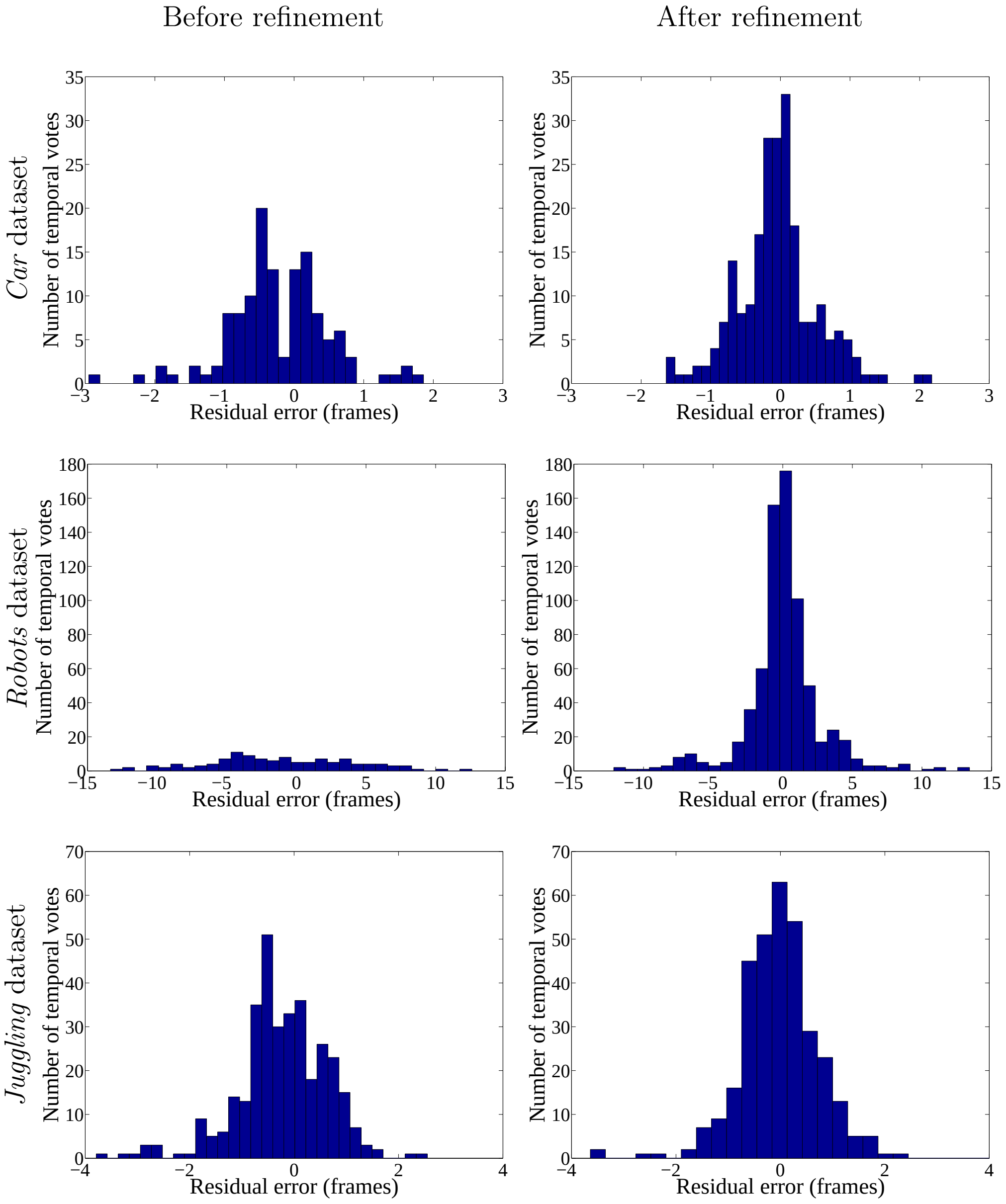


Figure 5.8: Distribution of distances of inlier votes from the reconstructed timeline. **Left column:** Distribution before the timeline refinement stage. **Right column:** Distribution after the refinement stage. Note that the updated epipolar geometry and updated timeline parameters reduce the distance between inliers and the timeline and cause more votes to be labeled as inliers.

5.1.4 Three-view *Soccer* Dataset

As a final experiment, we applied our technique to three video clips extracted from a single MPEG-compressed TV broadcast of a soccer match (FIFA, 2002). The clips were replays of the same goal filmed from three distinct viewpoints (Figure 5.9). Each sequence contained a significant panning motion to maintain the moving players within the field of view. To ensure that the pairwise fundamental matrices remained constant for all frames, we stabilized each sequence by computing the frame-to-frame homography using Brown and Lowe's system (Brown and Lowe, 2003). We used the WSL tracker to track the same player in each sequence, thereby obtaining one feature trajectory per camera. WSL was initialized manually in the first frame of each sequence. Even though it was able to track the chosen player for most frames, the player's small size and jitter artifacts caused by the video's poor quality resulted in noisy measurements of his location. These measurements were given as input to the basic timeline reconstruction algorithm with $\epsilon = 1.5$ and no timeline refinement.

Since this dataset contained $N = 3$ views, the timeline is a 3D line with 3-vectors as its coefficients (see Eq. (3.2) and Figures 5.10(b) and 5.10(c)). To evaluate the timeline's accuracy in the absence of ground-truth information, we attempted to estimate the ground-truth alignment by visual inspection: we identified three easily-distinguishable events (e.g., a player stepping on a field line, as in Figure 5.9) and recorded the frame where each event occurred in each sequence. These frames were used as "ground-truth" event times for each camera. To evaluate the timeline's accuracy, we used it to predict the event times in cameras 1 and 2 from the ground-truth time in camera 3.

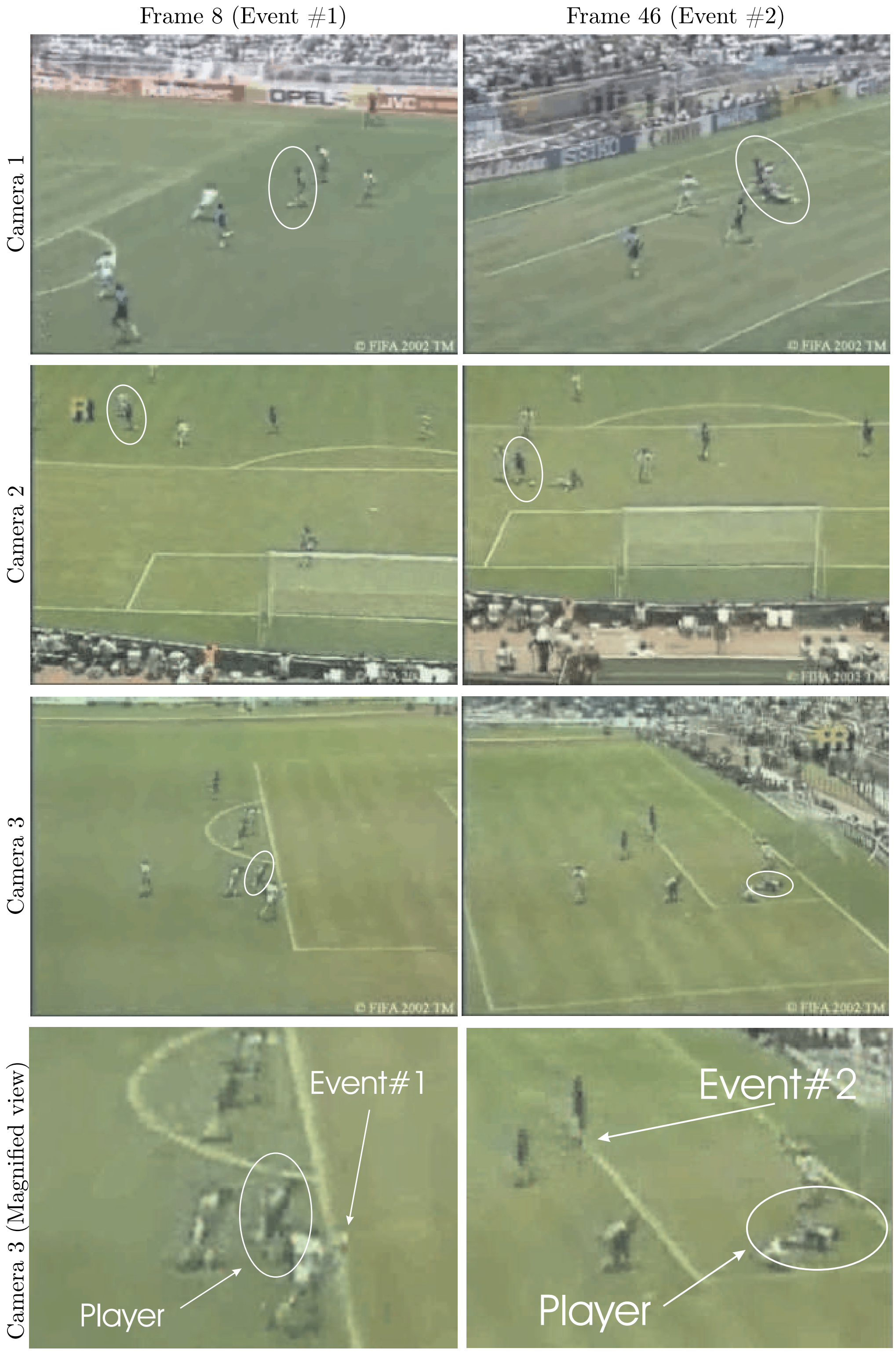
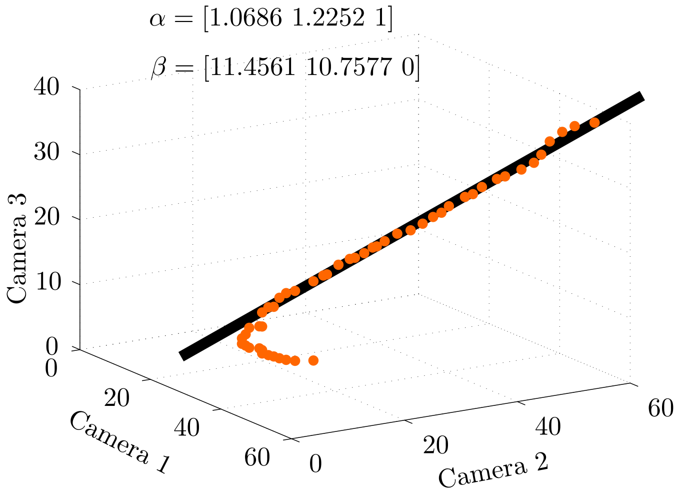
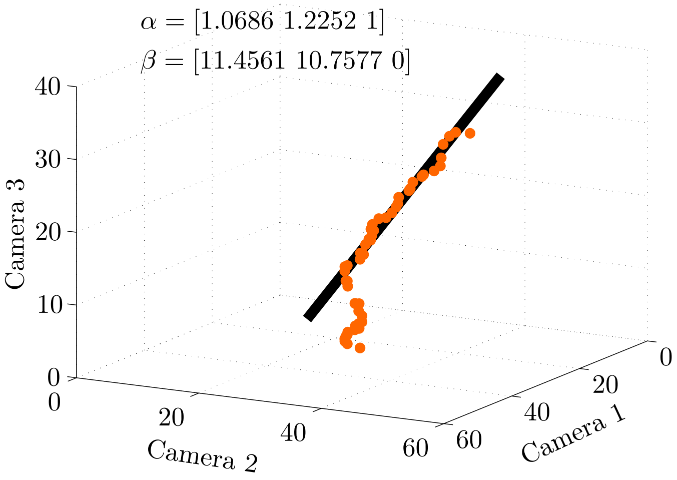


Figure 5.9: Two representative frames (8, 46) from the cameras 1, 2 and 3, of the three-view *Soccer* dataset. This dataset, along with more experimental results and videos, are available at *http://www.dcc.ufmg.br/˜cardeal/research/timeline/* .



(a)



(b)

Figure 5.10: (a),(b) Two views of the 3D voting space and 3D timeline computed for the *Soccer* dataset.

The minimum difference between the predictions and the ground-truth times across all three events was 0.22 frames in camera 1 and 0.86 frames in camera 2; the maximum difference was 1.66 and 1.33 frames, respectively. This confirms that the sequences were aligned quite well (see Figure 5.11), despite the low quality of the videos and their unequal frame rates.

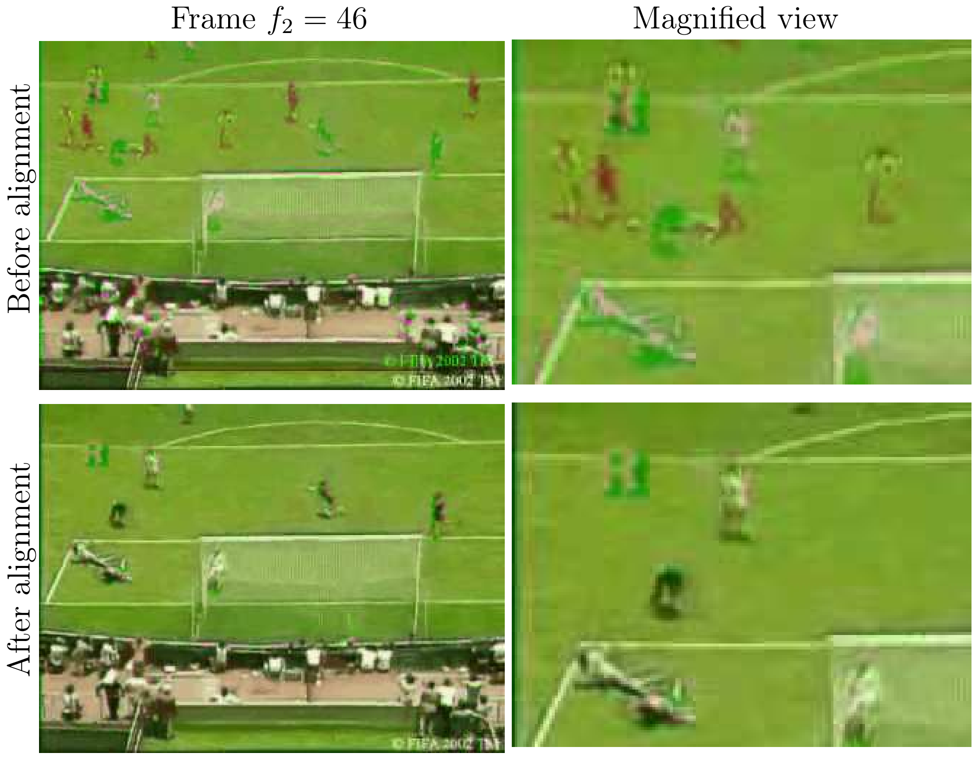


Figure 5.11: **Before alignment images** were created by superimposing the green band of a frame f_2 with the red and blue bands of frame $f_1^* = (f_2 - \beta^*)/\alpha^*$ using ground truth timeline coefficients α^* and β^*. **After alignment images** were created by replacing the green band of the images above them with that of frame $f_1 = (f_2 - \beta)/\alpha$, with α, β computed by our algorithm. For both types of images, deviations from the ground-truth alignment cause "double exposure" artifacts (i.e., when $f_1^* \neq f_2$ or $f_1^* \neq f_1$, respectively).

5.2 Synthetic Sequences

In the second phase of our experiments, we developed a software for simulating the dynamics of 3D particles with independent motion and their visualizations from multiple viewpoints. Through the use of synthetic data, the values of some key parameters, such as the number of tracked features, could be controlled and their impacts in the accuracy, stability and scalability

Cameras	Intrinsic parameters	Extrinsic parameters
C_1	$\begin{bmatrix} 675.00 & 0 & 438.69 \\ 0 & 674.61 & 260.08 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.71 & 0.01 & 0.70 & -1237.20 \\ -0.04 & -1.00 & 0.06 & 2.57 \\ 0.70 & -0.07 & -0.71 & 2636.10 \end{bmatrix}$
C_2	$\begin{bmatrix} 1835.30 & 0 & 352.39 \\ 0 & 1834.20 & 220.85 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.78 & -0.28 & 0.55 & -222.31 \\ 0.12 & -0.81 & -0.58 & 115.09 \\ 0.61 & 0.52 & -0.60 & 10130.00 \end{bmatrix}$

Table 5.1: Matrices of intrinsic and extrinsic parameters for the cameras used during the simulation of the artificial scene.

of our approach could be carefully analyzed. In particular, by using the developed simulator we answer in this section three fundamental questions:

1. What is the scalability of our method against an increasing number of tracked features?

2. How is the accuracy of our method affected by errors in the initial estimates of the fundamental matrices that capture the geometric relations between the views?

3. How does the noise level in the tracking system affect the accuracy of our technique?

5.2.1 Scene dynamics simulation

We consider in this analysis an artificial scene monitored by two calibrated cameras, whose intrinsic and extrinsic parameters are listed in Table 5.1 and are those of the real cameras used in our indoor experiments presented in Section 5.1, namely, an Hitachi KP-D50 Color CCD Camera ($C1$) and a Sony DCR-TRV320 Digital Camcorder (C_2). Those cameras were positioned according to the scheme illustrated in Figure 5.12(a).

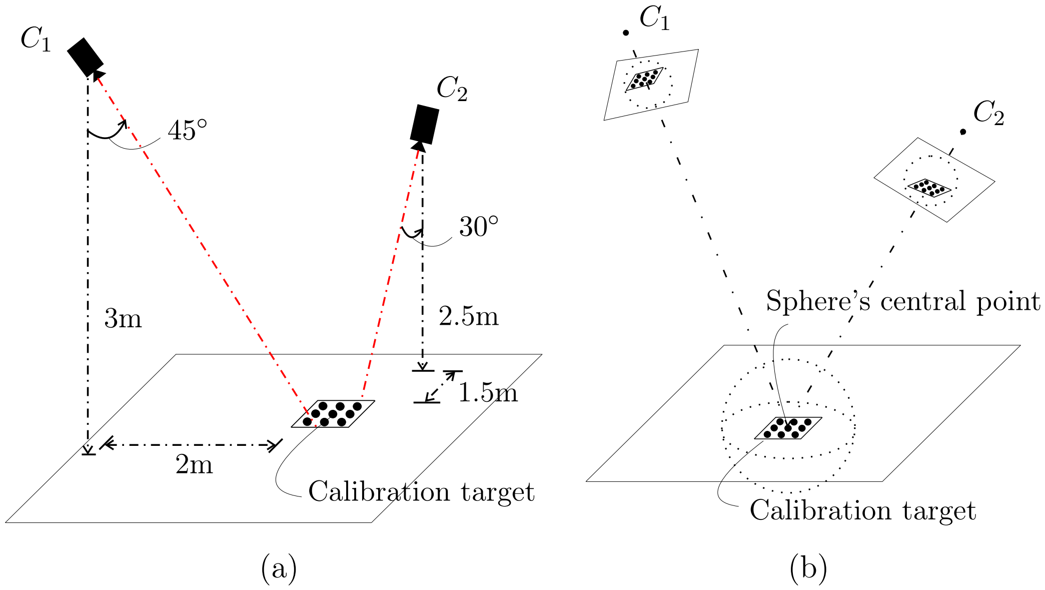


Figure 5.12: Simulation of scene dynamics. (a) Experimental setup. (b) All features start in random positions uniformly distributed within the illustrated sphere, which is viewed from both cameras.

Random object trajectories were generated in the world coordinate system and projected in the image planes by using the afore-mentioned intrinsic and extrinsic parameters. All features were started in random positions uniformly distributed within a sphere that was visible from both cameras, as illustrated in Figure 5.12(b). Note that the sphere's central point was defined by the center of the black circle located in the middle of a planar calibration target that was positioned in the 3D world, so that the projections of the central point of that circle were located near the centers of the cameras' CCDs. The sphere's radius was determined empirically in order to maximize its projections in both cameras, subject to the constraint that such projections should be entirely visible.

In our simulation, each feature moves for l frames, where l is drawn from a uniform distribution in the interval $[1, 2L]$. Specifically, we consider that

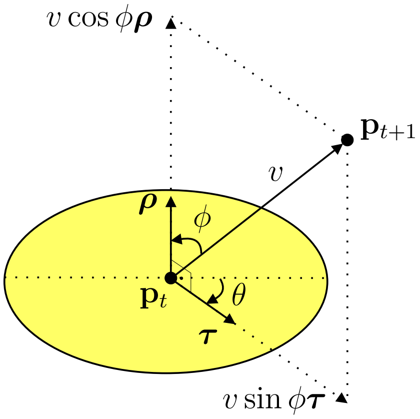


Figure 5.13: Feature dynamics.

both cameras acquire image sequences of the same lenght (256 frames) and that the average feature lifetime L is 128 frames. By defining a potentially distinct lifetime for each feature, we simulate situations where the tracker may lose some features, either because they actually became occluded or go out of bounds, or because the computation fails for one reason or another. As it is frequently desired to maintain a certain number of features, every time a feature dies, a new one is started in a random position within the volume visible from both cameras.

The features move with the following dynamics:

$$\mathbf{p}_{t+1} = \mathbf{p}_t + v \cos \phi \boldsymbol{\rho} + v \sin \phi \boldsymbol{\tau}, \tag{5.1}$$

which is illustrated in Figure 5.13 and whose notation is described below:

- $\mathbf{p}_t$ and $\mathbf{p}_{t+1}$ are the current frame and next frame positions, respectively.
- v is the magnitude of the feature displacement in the 3D world (in *cm*), simulated as additive gaussian noise with zero mean and standard deviation V. In the current simulation, we defined $V = \pm 2.5$cm, resulting in an average feature displacement of about 2 pixels in the image planes of both cameras.

- ϕ is an angular variation in the direction of movement of the feature in the 3D world (in radians), simulated as additive gaussian noise with mean zero and standard deviation A. Specifically, we defined $A = 0.09$ radians, that is, about 5 degrees.
- $\boldsymbol{\rho}$ is a unit direction vector along the current direction of movement.
- $\boldsymbol{\tau}$ is a unit direction vector perpendicular to $\boldsymbol{\rho}$, given by the cross product between $\boldsymbol{\rho}$ and a unit direction vector uniformly sampled on the sphere.

Regarding the video sequences obtained by the visualization process performed by the cameras, we defined the following "ground-truth" affine transformation for modelling their temporal misalignment:

$$f_2 = f_1 - 32, \tag{5.2}$$

that is, we have cameras with the same frame rate ($\alpha = 1$) and whose corresponding video sequences have a temporal misalignment of 32 frames ($\beta = -32$). Importantly, the quality of the temporal alignments estimated by our method as well as its computational cost are invariant to the magnitude of the temporal shift between the sequences, since our approach derives alignment constraints by matching instantaneous positions of features in one sequence against entire feature trajectories in the other sequences. Therefore, we have arbitrarily defined a temporal misalignment of 32 frames, which represents approximately the average value of most of the temporal shifts of the real-world sequences presented in Section 5.1. Given that both cameras used in this experiment work at a frame rate of 30 frames per second, we note that this temporal shift simulates cameras that were activated at distinct time instants with an interval of about 1 second.

5.2.2 Error measurements and experiments' description

In the current analysis, we use the average temporal alignment error as our basic measurement for evaluating the accuracy, scalability and stability of our approach. Specifically, its value is given by the average of the absolute values of the differences between the temporal coordinates computed by the estimated timeline and the temporal coordinates computed by the "ground-truth" affine transformation in Equation (5.2). That is, if f_1 represents the temporal coordinate of the reference sequence, $t_g(f_1)$ represents its corresponding temporal coordinate computed by the affine transformation in Equation (5.2) and $t_e(f_1)$ is its corresponding temporal coordinate computed by using the timeline estimated by our method, the average temporal alignment error ε_t is given by:

$$\varepsilon_t = \frac{1}{256} \sum_{f_1=0}^{255} |t_e(f_1) - t_g(f_1)| = \frac{1}{256} \sum_{f_1=0}^{255} |(\alpha - 1)f_1 + \beta + 32| \,. \tag{5.3}$$

To measure the error of the fundamental matrix, we used the same background features that were used to compute its initial estimate in the real scene. Specifically, this error was measured as the average of the distances between each background feature projection in the image plane of the reference camera (C_1) and its corresponding epipolar line. That is, if e_i represents the distance between a static feature i and its corresponding epipolar line, the error ε_f (in pixels) of the fundamental matrix is given by:

$$\varepsilon_f = \frac{1}{m} \sum_{i=1}^{m} e_i, \tag{5.4}$$

where m is the total number of background features.

In order to simulate the error in the cameras' epipolar geometry, new fundamental matrices with pre-determined errors (in pixels) were generated by adding a small multiple $\Delta_{\mathcal{F}}$ of a 3×3 unit matrix $\mathcal{J}$ to the original fundamental matrix $\mathcal{F}$ estimated with the normalized eight-point algorithm. That is, we added the term $\Delta_{\mathcal{F}} = \delta_{\mathcal{F}}\mathcal{J}$ to $\mathcal{F}$, where $\delta_{\mathcal{F}}$ simulates a gaussian noise with mean zero and standard deviation $1e - 5$. This operation was repeated until the new fundamental matrix had the desired error.

On the other hand, to simulate distinct noise levels in the tracker, we considered the following model:

$$\begin{aligned} x_n &= x_o + d \cos\varphi, && (5.5) \\ y_n &= y_o + d \sin\varphi, && (5.6) \end{aligned}$$

where:

- (x_o, y_o) represents the original coordinates of a feature.
- (x_n, y_n) represents the the new corrupted coordinates of a feature.
- d represents the magnitude of the feature displacement, simulated as additive gaussian noise with zero mean and standard deviation R.
- φ defines the direction of the feature displacement in the image plane and is drawn from a uniform distribution in the interval $[0, 2\pi]$.

Particularly, we controlled the variation of the tracker's noise level in our experiments by defining specific values for the standard deviation R of the random variable d described above.

Using the above-described mechanisms for defining and measuring the errors of the estimated temporal alignments, of the cameras' epipolar geometry and of the feature positions estimated by the tracker, we performed the following two groups of experiments:

1. **Varying number of features and tracker's error with a fixed fundamental matrix's error of 2 pixels** ($\varepsilon_f = 2$). In this case, for each tuple (ε_f, R, k), where $R \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ and $k \in \{1, 2, 4, 8, 16, 32\}$, we simulated 100 distinct 256-frame 3D sequences, computed their corresponding voting spaces and measured the percentages of timelines that lead to average temporal alignment errors smaller than or equal to 1, 2 and 5 frame(s), before and after the timeline refinement stage.

2. **Varying number of features and fundamental matrix's error for a tracker with a gaussian noise that has a fixed standard deviation of ±2 pixels** ($R = 2$). Similarly to the previous group of experiments, for each tuple (ε_f, R, k), where $\varepsilon_f \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ and $k \in \{1, 2, 4, 8, 16, 32\}$, we simulated 100 distinct 256-frame 3D sequences, computed their voting spaces and measured the percentages of timelines that lead to average temporal alignment errors smaller than or equal to 1, 2 and 5 frame(s).

Since a fundamental matrix with an average error of 2 pixels represented the worst case in the real-world sequences presented in Section 5.1, we decided to use that value in our first group of experiments above, in order to guarantee a realistic perception about the scalability of our approach in cases where the computation of very accurate fundamental matrices is impossible. Similarly, a tracker with a standard deviation of ±2 pixels was used in the second group of experiments, since that value represents roughly the noise level observed in some trackers used in our real-world sequences.

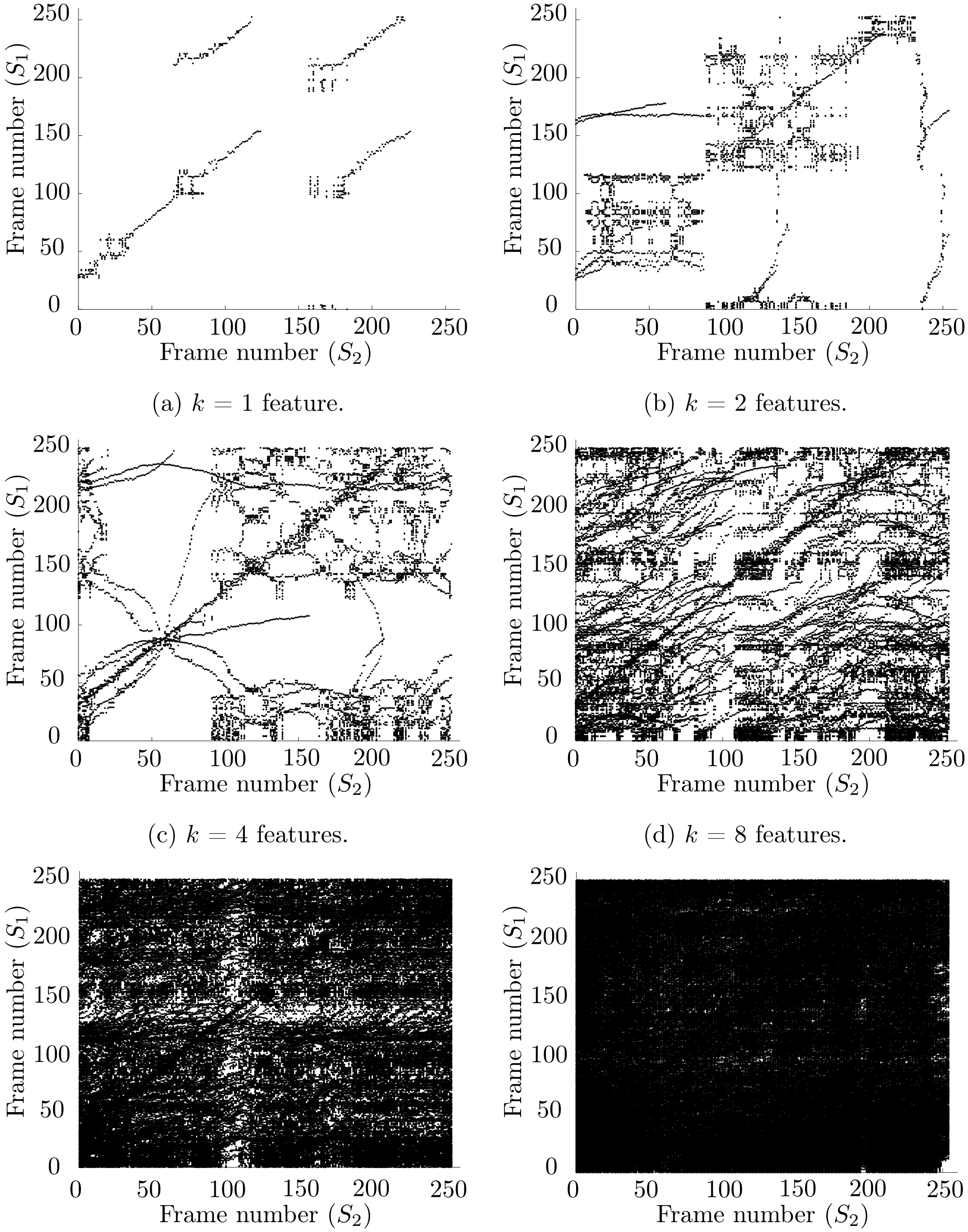
By measuring the percentages of timelines that lead to average temporal alignment errors smaller than or equal to 1, 2 and 5 frame(s), we illustrate the applicability of our approach in scenarios where we need timelines with highly accurate parameters ($\varepsilon_t \leq 1$ frame) or in less challenging situations where lower accuracies are allowed ($\varepsilon_t \leq 2$ frames or even $\varepsilon_t \leq 5$ frames).

The values used in all experiments for the RANSAC parameters were: $p = 0.99$, $r = 0.05$ and $\epsilon = 0.5$. According to Equation (3.3), those values lead to a RANSAC execution of 1840 iterations. Importantly, they represent conservative values that were defined in order to maximize the accuracy of the timeline's parameters, even though they also result in a lower efficiency for our method. Moreover, in order to ensure the acquisition of accurate results, we restricted the search of the temporal alignment algorithm for a timeline whose angular coefficient α satisfies: $0.2 \leq \alpha \leq 5$, that is, we used an *a-priori* information that the video sequences have frame rates that differ by at most a factor of 5.

5.2.3 Evaluation of the experimental results

As a first step, we evaluated the scalability of our temporal alignment and refinement methods against an increasing of the number of moving features per frame. Specifically, we considered scenes with $k = 1$, 2, 4, 8, 16 and 32 feature(s). In Figure 5.14 we illustrate some examples of voting spaces for those values of features, where the fundamental matrix has an average error of 2 pixels and the tracker is corrupted by a gaussian noise with zero mean and standard deviation of ±2 pixels.

Note that the larger is the number of features, the denser are the voting spaces. In fact, by increasing the number of features, the ratio between inliers and outliers decreases visibly. Figure 5.15 quantifies this visual perception by presenting the percentage of inliers for each voting space of Figure 5.14, where a specific vote is regarded as an inlier if its temporal coordinates f_1 and f_2 satisfy $f_2 - f_1 + 32 \leq 1$ frame.



(a) $k = 1$ feature.

(b) $k = 2$ features.

(c) $k = 4$ features.

(d) $k = 8$ features.

(e) $k = 16$ features.

(f) $k = 32$ features.

Figure 5.14: Examples of voting spaces for $k = 1$, 2, 4, 8, 16 and 32 features. In all these cases, the fundamental matrix has an average error of 2 pixels ($\varepsilon_f = 2$) and the tracker is corrupted by a gaussian noise with mean zero and standard deviation of ±2 pixels ($R = 2$).

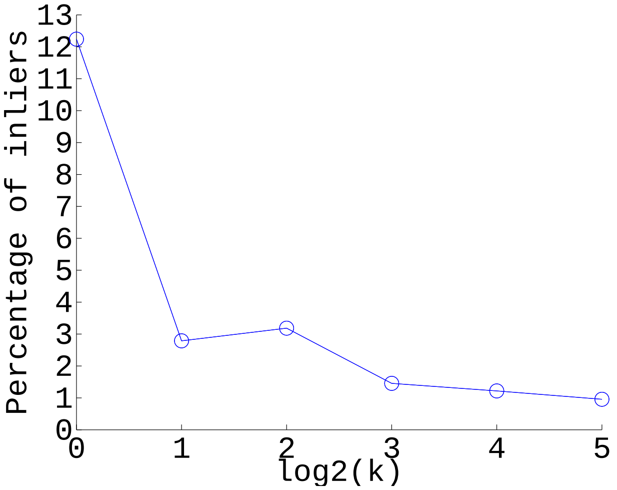


Figure 5.15: Percentage of inliers for each voting space of Figure 5.14, where k represents the number of tracked features (k = 1, 2, 4, 8, 16 and 32). A specific vote is considered an inlier if its coordinates f_1 and f_2 satisfy $f_2 - f_1 + 32 \leq 1$ frame.

The voting spaces for 16 and 32 features represent especially challenging cases, where it is impossible to identify visually the actual timeline. According to Figure 5.15, we have less than 2% of inliers in both cases.

Consider now Figures 5.16 to 5.25. Those figures illustrate the percentages of timelines that lead to average temporal alignment errors smaller than or equal to 1, 2 and 5 frame(s), as a function of the number of tracked features. In particular, Figures 5.16 to 5.20 illustrate scenarios with distinct tracker's errors, but that have in common a fundamental matrix with an average error of 2 pixels. On the other hand, Figures 5.21 to 5.25 illustrates situations where different fundamental matrix's errors are considered and a tracker that is corrupted by a gaussian noise with standard deviation of ±2 pixels is used.

By observing the curves that illustrate the percentages of timelines that lead to average temporal alignment errors smaller than or equal to 1, 2 and

5 frame(s) in all those figures, both before and after the refinement stage, we note that although sets containing from 2 to 16 features bring similar results when the tracker's noise level and the fundamental matrix's error have small values, as illustrated in Figures 5.16(a)-(f) and 5.21(a)-(f), our approach tends to be more accurate when sets of 4 features are used, especially in scenarios where the tracking system and the initial estimate of the cameras's epipolar geometry have errors with more significative magnitudes, as illustrated in Figure 5.20(d)-(f) where the tracker is corrupted by a noise with standard deviation of ±10 pixels, and in Figures 5.22(a)-(f) and 5.23(a)-(f) for fundamental matrix's errors varying from 3 to 6 pixels.

Specifically, if we consider sets of about 4 features, a fundamental matrix's error of about 2 pixels and applications that need timelines whose temporal alignment errors are smaller than or equal to 1 frame, we note from Figures 5.16 to 5.20 that the percentage of solutions computed by our method that satisfy this errors constraint vary from 100%, as illustrated in Figure 5.16(a) for a tracker with a low noise level ($R = 1$), to about 30%, as illustrated in Figure 5.20(d) for a case where the tracker was severely corrupted by noise ($R = 10$). On the other hand, by fixing now the tracker's noise level ($R = 2$) and varying the fundamental matrix's error, we observe from Figures 5.21 to 5.25 that when sets with 4 features are used, the percentage of timelines whose temporal alignment errors are smaller than or equal to 1 frame vary from 90%, as shown in Figure 5.21(a) for a fundamental matrix's error of 1 pixel ($\varepsilon_f = 1$), to about 5%, as illustrated in Figure 5.25(d) for a fundamental matrix's error of 10 pixels ($\varepsilon_f = 10$). Therefore, note that the accuracy of our approach, in special the accuracy of the temporal alignment algorithm, is much more negatively affected by an increase in the error of the fundamental matrix than by an equivalent increase of the tracker's noise level.

Importantly, by assuming less challenging upper bounds of 2 and 5 frames for the temporal alignment error, we observe that the percentage of timelines computed by our method that must satisfy these error constraints increases significantly. In fact, by using a set with 4 features and considering a temporal alignment error of at most 2 frames, our method succeeds in 65% of the total number of trials (after the refinement stage) even when a tracker with a high noise level ($R = 10$) is used, as shown in Figure 5.20(e), and in 55% of the trials (after the refinement stage) when a highly inaccurate fundamental matrix ($\varepsilon_f = 6$) is computed, as illustrated in Figure 5.23(e). The results are even better in cases where the temporal alignment error may be up to 5 frames as, for example, in Figures 5.20(f) and 5.25(f), where our approach succeeded in about 100% and 50% of the total number of trials, respectively, even though the tracker's noise level ($R = 10$) and the fundamental matrix's error ($\varepsilon_f = 10$) had large values.

Additionally, note from Figures 5.16 to 5.25 that the accuracy of the timeline's parameters may decrease significantly when feature sets too small (e.g., 1 feature) or too large (e.g., 32 features) are used. This behavior of our methodology is explained by the direct relation between the complexity of the voting space and the number of tracked features considered. When more features are added, the number of outliers increases faster than the number of inliers, as illustrated in Figures 5.14 and 5.15. Therefore, with a higher amount of spurious information in the voting space due to the higher number of features, our approach may compute timelines whose parameters are more inaccurate. On the other hand, the smaller is the set of features the smaller is the number of votes in the voting space. In this case, an insufficient number of inliers may also result in the estimation of inaccurate parameters for the timeline that models the temporal alignment between the sequences.

However, Figure 5.16(a)-(c) suggests that in cases where trackers with low noise levels are used, an increase in the number of tracked features does not cause much impact in the accuracy of the timeline's parameters estimated, especially before the application of the refinement technique. Note in Figure 5.16(a), for example, the small variations in the percentage of correct timelines before the refinement stage, when the number of features increases.

To illustrate the variation of the values of the timeline's parameters when the number of features increases, we present Figures 5.26 to 5.28. For all the cases illustrated in those figures, we consider that the average error of the fundamental matrix and the standard deviation of the gaussian noise added to the tracker are 2 pixels. Moreover, only the parameters of timelines that lead to an average temporal alignment error smaller than or equal to 1 frame were considered. We note that for all values of features illustrated in those figures, the refinement stage played an important role. After its application, the average values of the timeline's parameters became closer to the ground-truth parameters and their corresponding variances decreased significantly. This fact is specially noticeable in the cases of sets with 4 and 16 features, which are illustrated in Figures 5.27(a) and 5.28(a), respectively.

Now, consider Figures 5.29 to 5.34. Essentially, those figures provide the same information already presented in Figures 5.16 to 5.25. However, differently from those previous figures, Figures 5.29 to 5.34 illustrate the percentage of timelines that lead to average temporal alignment errors smaller than or equal to 1, 2 and 5 frame(s), as a function of the standard deviation of the gaussian noise added to the tracker (Figures 5.29 to 5.31) and as a function of the error in the initial estimate of the fundamental matrix (Figures 5.32 to 5.34). Therefore, by analyzing Figures 5.29 to 5.34 together with Figures 5.16 to 5.25, better considerations may be performed about the impacts in the accuracy and stability of our results caused by errors in the cameras' epipolar geometry and in the tracking system.

By observing the curves in Figures 5.29 to 5.34 that illustrate the percentages of timelines that lead to average temporal alignment errors smaller than or equal to 1, 2 and 5 frame(s), both before and after the application of the refinement technique, we note that the higher is the tracker's noise level and/or the fundamental matrix's error the higher is the tendency of the temporal alignment and refinement methodologies to present lower accuracies. This already expected behavior of our approach is more critical when more features are considered, as illustrated in Figures 5.31(a)-(f) and 5.34(a)-(f) for sets with 16 and 32 features.

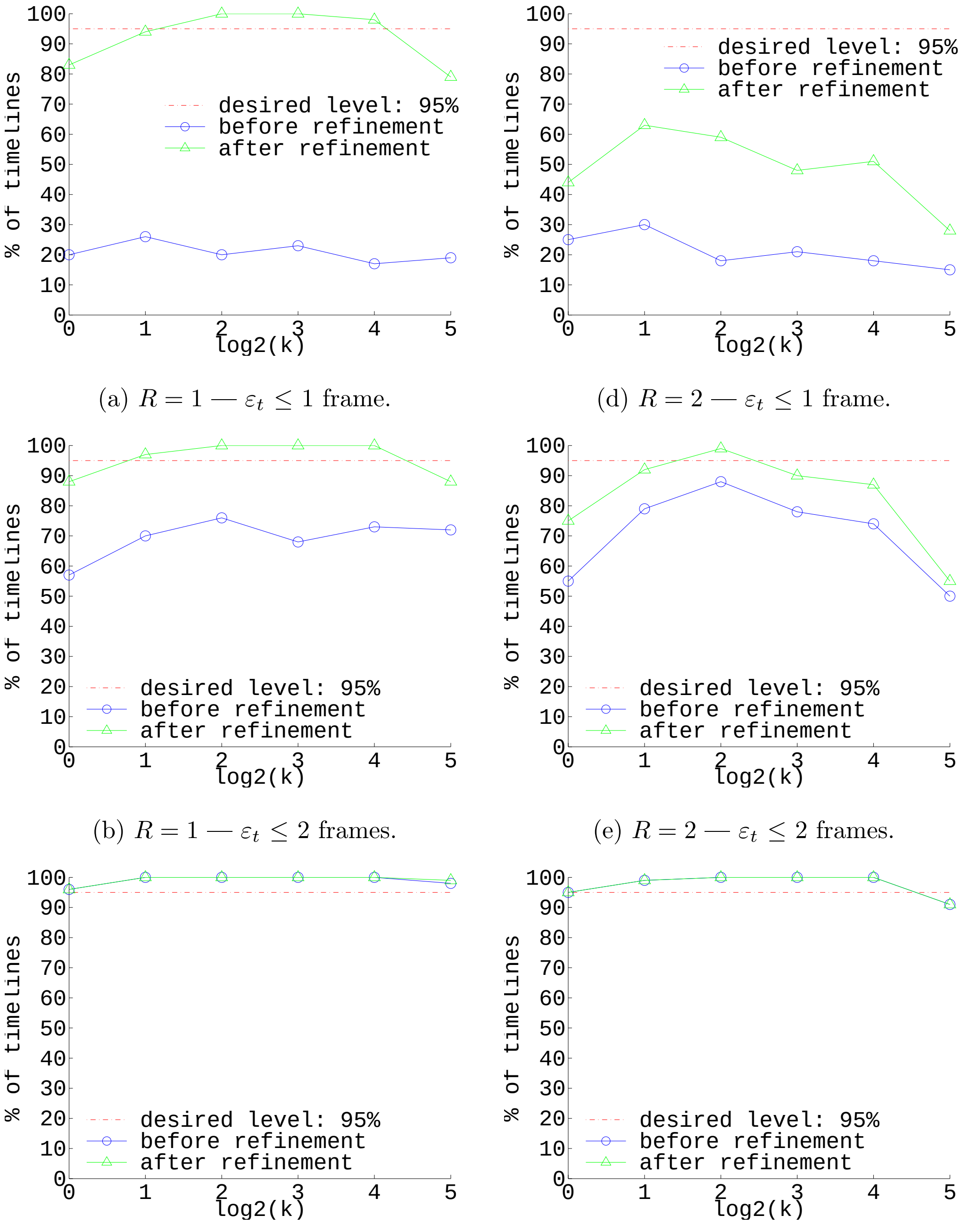
Two main reasons explain this degradation in the accuracy of our results due to increases in the tracker's noise level and in the error of the cameras' epipolar geometry. The first reason relates to the fact that the higher are those errors the lower is the information quality regarding the actual correspondences between temporal coordinates of feature positions in both image planes. That is, by considering the binary representation of a candidate point in the voting space, those errors produce an effect that is equivalent to induce a lost of some of its less significative bits. In this context, potential inliers are shifted from their actual positions in the voting space, where the magnitudes of those shifts are proportional to the errors' magnitudes. This behavior may affect the estimation process performed by RANSAC resulting in timelines with inaccurate parameters.

The second reason is similar to the one appointed previously regarding increases in the number of features. That is, when trackers and fundamental matrices with higher errors are used, we observe a significative increase of spurious information in the voting spaces. Consequently, this higher number of outliers may affect negatively the effectiveness of RANSAC and lead our approach to compute timelines with inaccurate parameters.

Importantly, the more inaccurate are the timeline's parameters the harder is for the optimization task formulated in the refinement stage to converge to their actual values. This last fact explains the observation from Figures 5.29 to 5.34 that the higher is the tracker's noise level and the fundamental matrix's error the lower is the improvement brought by the application of the refinement technique. Consider, for instance, Figure 5.29(a) where we have only 1 feature in the scene. When the standard deviation of the tracker's noise increases and, consequently, timelines with less accurate parameters are estimated due to the reasons afore-mentioned, we note that the refinement technique has its effectiveness hardly affected. Specifically, Figure 5.29(a) suggests that for trackers with standard deviations larger than or equal to 6 pixels, our refinement method should not be used.

By observing Figures 5.29 to 5.31 and establishing that our approach should ideally estimate timelines with average temporal alignment errors smaller than or equal to 1 frame in 95% of the total number of trials, we note that for a fundamental matrix's error of about 2 pixels, that goal is achieved only when trackers with low noise levels (standard deviation of about 1 pixel) are used, as illustrated in Figures 5.30(a),(d) and 5.31(a) for sets containing 4, 8 and 16 tracked features.

Finally, we note from Figures 5.29(a),(d), 5.30(a),(d) and 5.31(a),(d) that the exclusive use of the temporal alignment technique is not appropriate when we need timelines whose temporal alignment errors must be smaller than or equal to 1 frame, since for any number of features and noise level considered, the percentage of timelines that satisfied that errors constraint before the refinement stage was usually smaller than 30%.



(a) $R = 1$ — $\varepsilon_t \leq 1$ frame.

(d) $R = 2$ — $\varepsilon_t \leq 1$ frame.

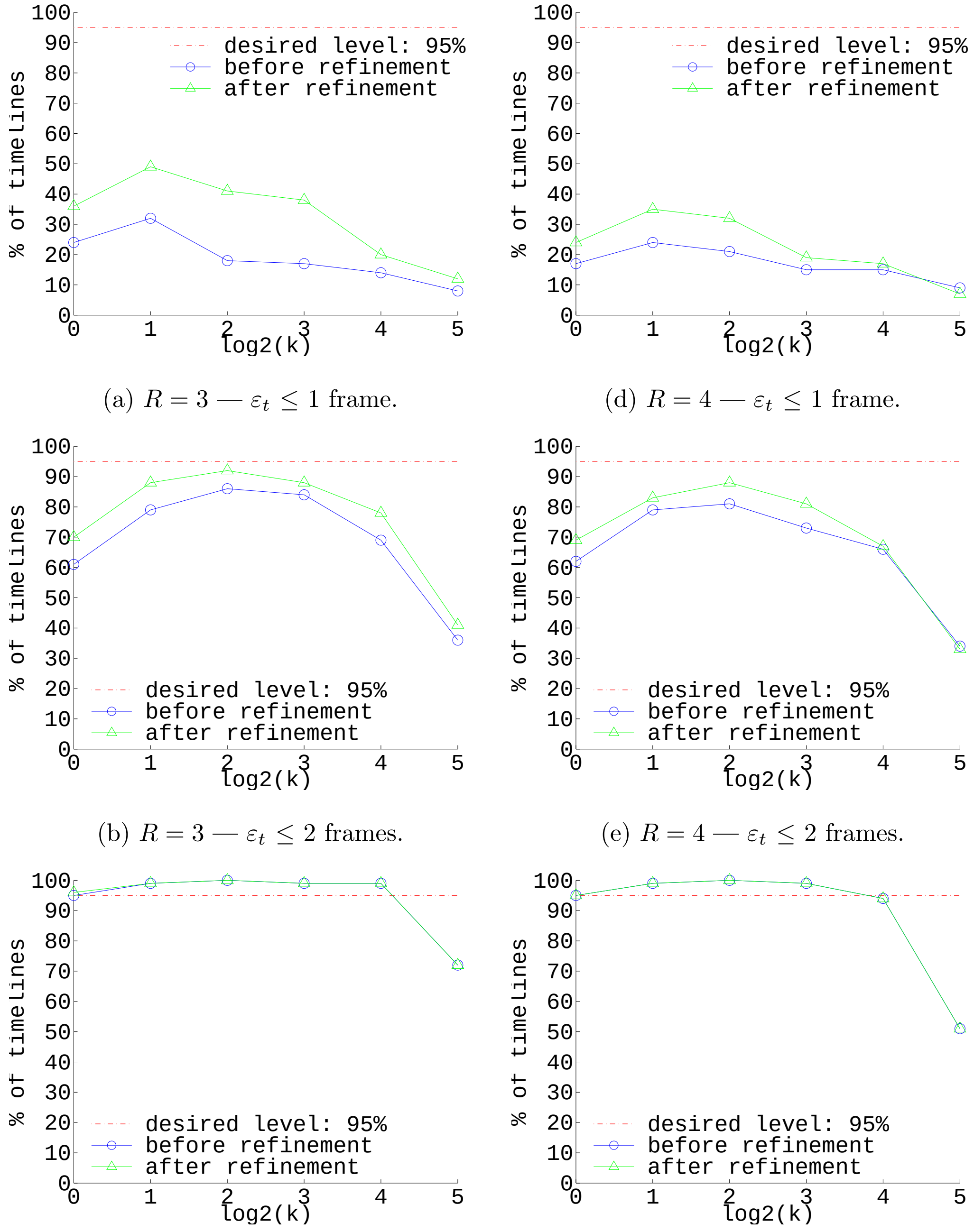
(b) $R = 1$ — $\varepsilon_t \leq 2$ frames.

(e) $R = 2$ — $\varepsilon_t \leq 2$ frames.

(c) $R = 1$ — $\varepsilon_t \leq 5$ frames.

(f) $R = 2$ — $\varepsilon_t \leq 5$ frames.

Figure 5.16: Percentages of timelines that lead to average temporal alignment errors smaller than or equal to 1, 2 and 5 frame(s), for $k = 1$, 2, 4, 8, 16 and 32 feature(s). (a), (b) and (c) Standard deviation R of the tracker's gaussian noise is ±1 pixel. (d), (e) and (f) Standard deviation R of the tracker's gaussian noise is ±2 pixels. In all cases, a fundamental matrix with an average error of 2 pixels is used ($\varepsilon_f = 2$).



(a) $R = 3$ — $\varepsilon_t \leq 1$ frame.

(d) $R = 4$ — $\varepsilon_t \leq 1$ frame.

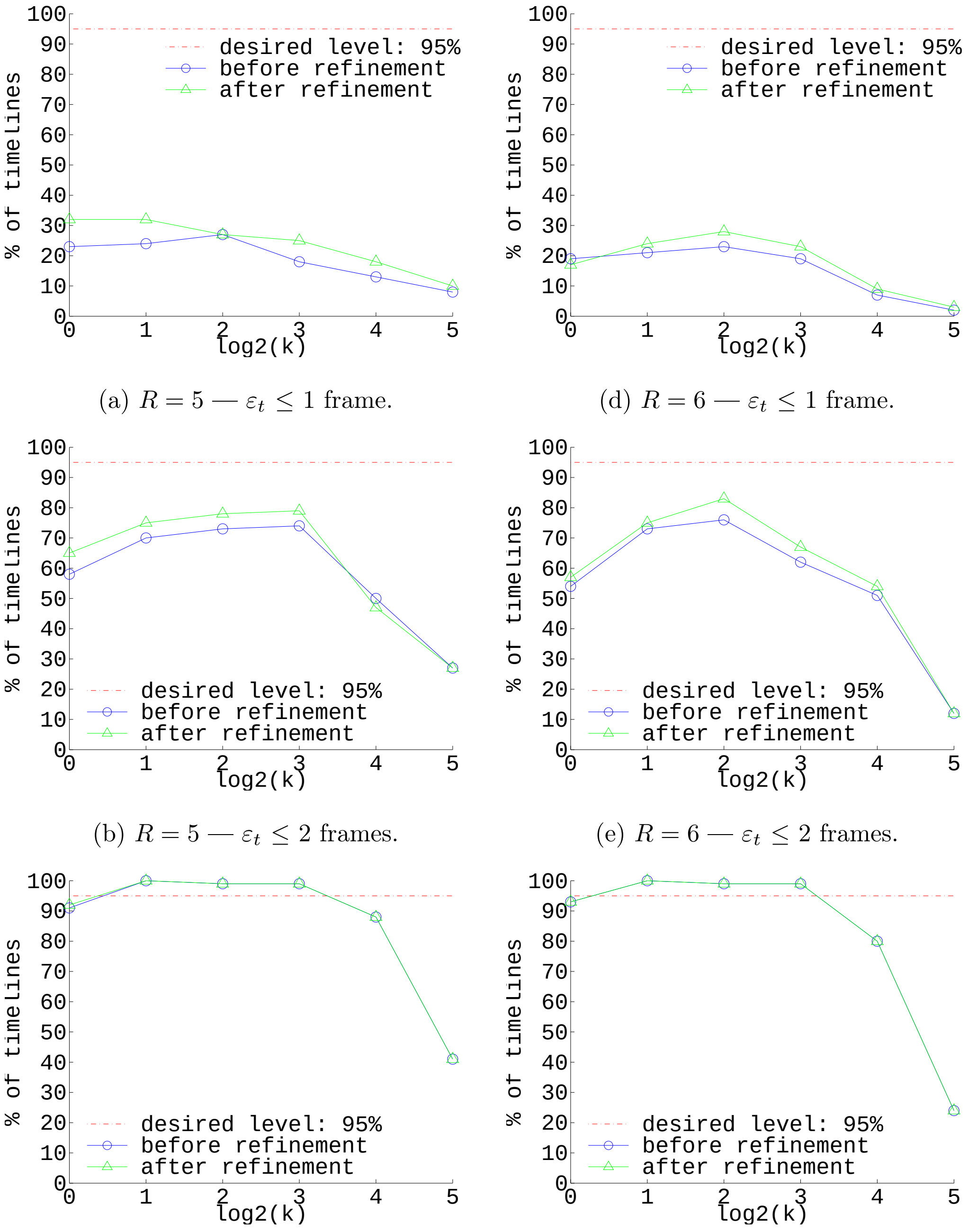
(b) $R = 3$ — $\varepsilon_t \leq 2$ frames.

(e) $R = 4$ — $\varepsilon_t \leq 2$ frames.

(c) $R = 3$ — $\varepsilon_t \leq 5$ frames.

(f) $R = 4$ — $\varepsilon_t \leq 5$ frames.

Figure 5.17: Percentages of timelines that lead to average temporal alignment errors smaller than or equal to 1, 2 and 5 frame(s), for k = 1, 2, 4, 8, 16 and 32 feature(s). (a), (b) and (c) Standard deviation R of the tracker's gaussian noise is ±3 pixels. (d), (e) and (f) Standard deviation R of the tracker's gaussian noise is ±4 pixels. In all cases, a fundamental matrix with an average error of 2 pixels is used ($\varepsilon_f = 2$).



(a) $R = 5$ — $\varepsilon_t \leq 1$ frame.

(d) $R = 6$ — $\varepsilon_t \leq 1$ frame.

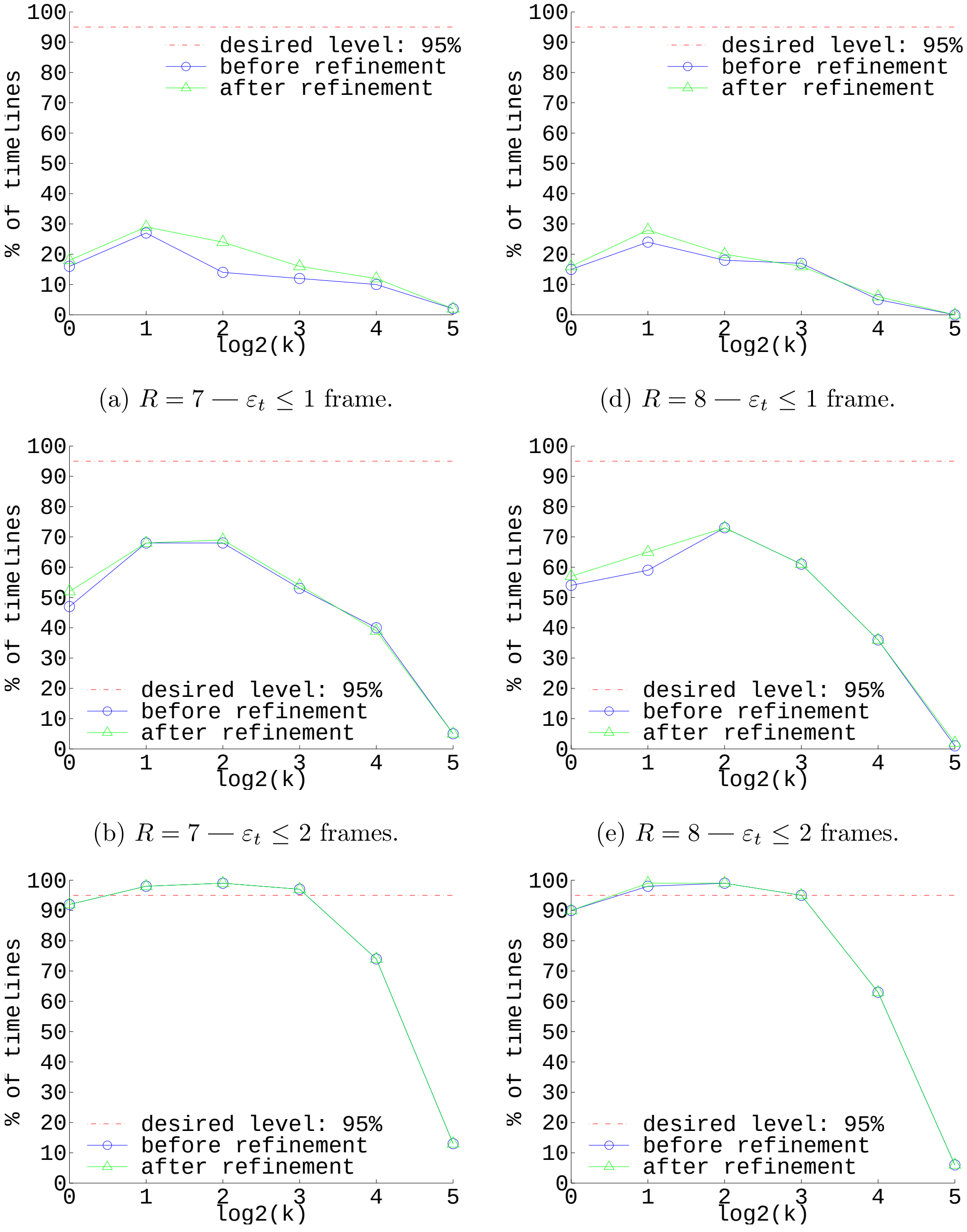
(b) $R = 5$ — $\varepsilon_t \leq 2$ frames.

(e) $R = 6$ — $\varepsilon_t \leq 2$ frames.

(c) $R = 5$ — $\varepsilon_t \leq 5$ frames.

(f) $R = 6$ — $\varepsilon_t \leq 5$ frames.

Figure 5.18: Percentages of timelines that lead to average temporal alignment errors smaller than or equal to 1, 2 and 5 frame(s), for $k = 1, 2, 4, 8, 16$ and 32 feature(s). (a), (b) and (c) Standard deviation R of the tracker's gaussian noise is ±5 pixels. (d), (e) and (f) Standard deviation R of the tracker's gaussian noise is ±6 pixels. In all cases, a fundamental matrix with an average error of 2 pixels is used ($\varepsilon_f = 2$).



(a) $R = 7$ — $\varepsilon_t \leq 1$ frame.

(d) $R = 8$ — $\varepsilon_t \leq 1$ frame.

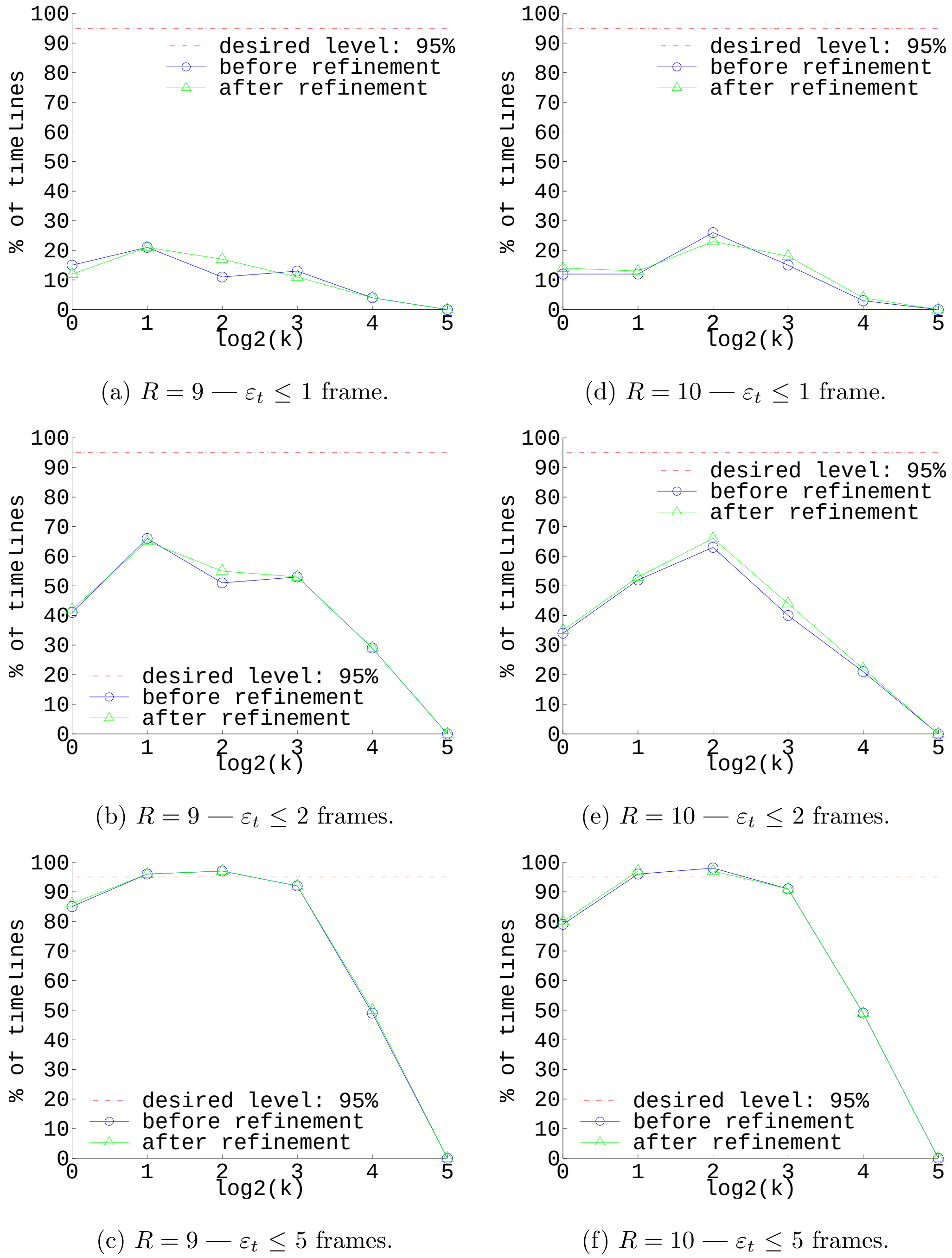
(b) $R = 7$ — $\varepsilon_t \leq 2$ frames.

(e) $R = 8$ — $\varepsilon_t \leq 2$ frames.

(c) $R = 7$ — $\varepsilon_t \leq 5$ frames.

(f) $R = 8$ — $\varepsilon_t \leq 5$ frames.

Figure 5.19: Percentages of timelines that lead to average temporal alignment errors smaller than or equal to 1, 2 and 5 frame(s), for $k = 1$, 2, 4, 8, 16 and 32 feature(s). (a), (b) and (c) Standard deviation R of the tracker's gaussian noise is ±7 pixels. (d), (e) and (f) Standard deviation R of the tracker's gaussian noise is ±8 pixels. In all cases, a fundamental matrix with an average error of 2 pixels is used ($\varepsilon_f = 2$).



(a) $R = 9$ — $\varepsilon_t \leq 1$ frame.

(b) $R = 9$ — $\varepsilon_t \leq 2$ frames.

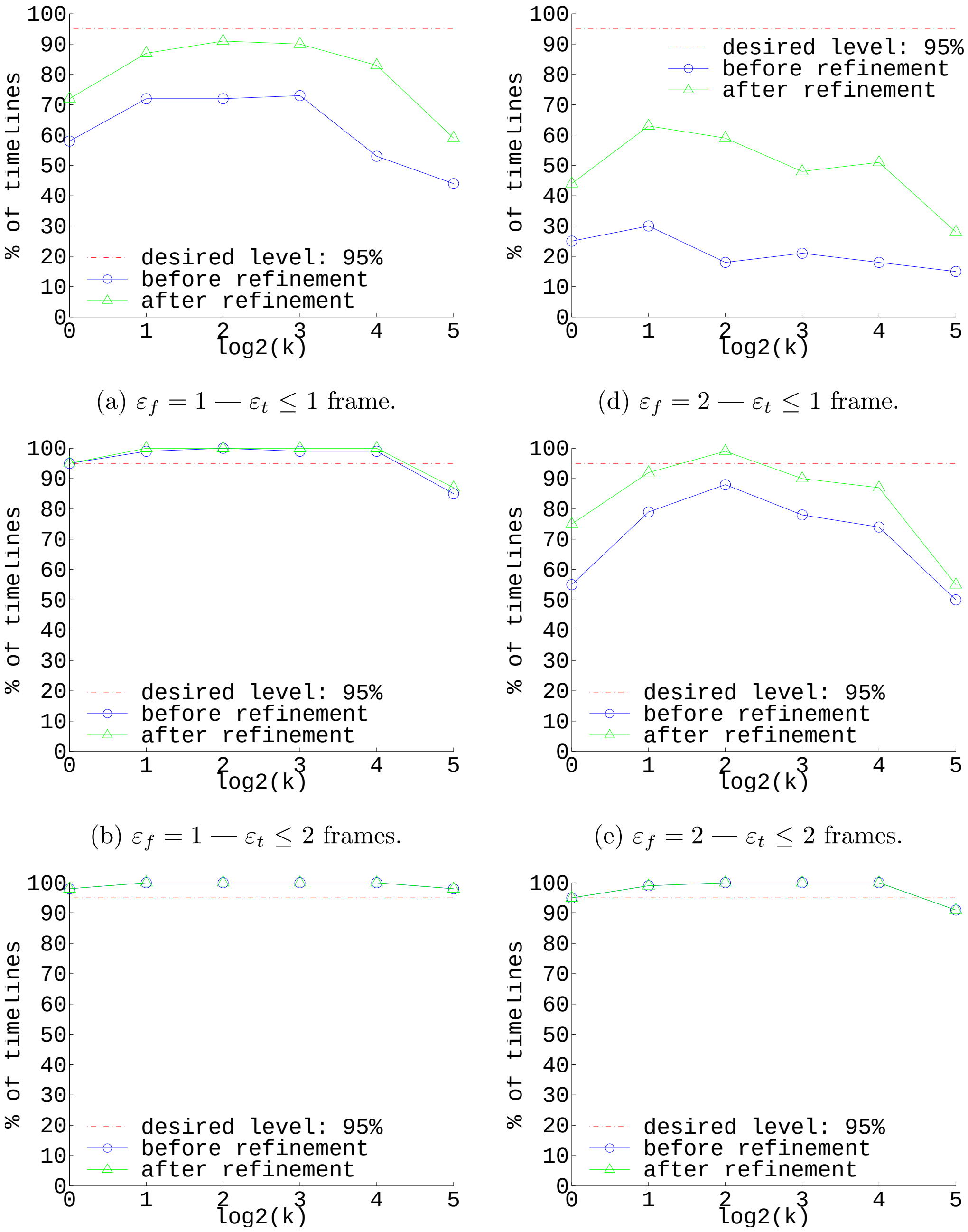
(c) $R = 9$ — $\varepsilon_t \leq 5$ frames.

(d) $R = 10$ — $\varepsilon_t \leq 1$ frame.

(e) $R = 10$ — $\varepsilon_t \leq 2$ frames.

(f) $R = 10$ — $\varepsilon_t \leq 5$ frames.

Figure 5.20: Percentages of timelines that lead to average temporal alignment errors smaller than or equal to 1, 2 and 5 frame(s), for $k = 1$, 2, 4, 8, 16 and 32 feature(s). (a), (b) and (c) Standard deviation R of the tracker's gaussian noise is ±9 pixels. (d), (e) and (f) Standard deviation R of the tracker's gaussian noise is ±10 pixels. In all cases, a fundamental matrix with an average error of 2 pixels is used ($\varepsilon_f = 2$).



(a) $\varepsilon_f = 1$ — $\varepsilon_t \leq 1$ frame.

(d) $\varepsilon_f = 2$ — $\varepsilon_t \leq 1$ frame.

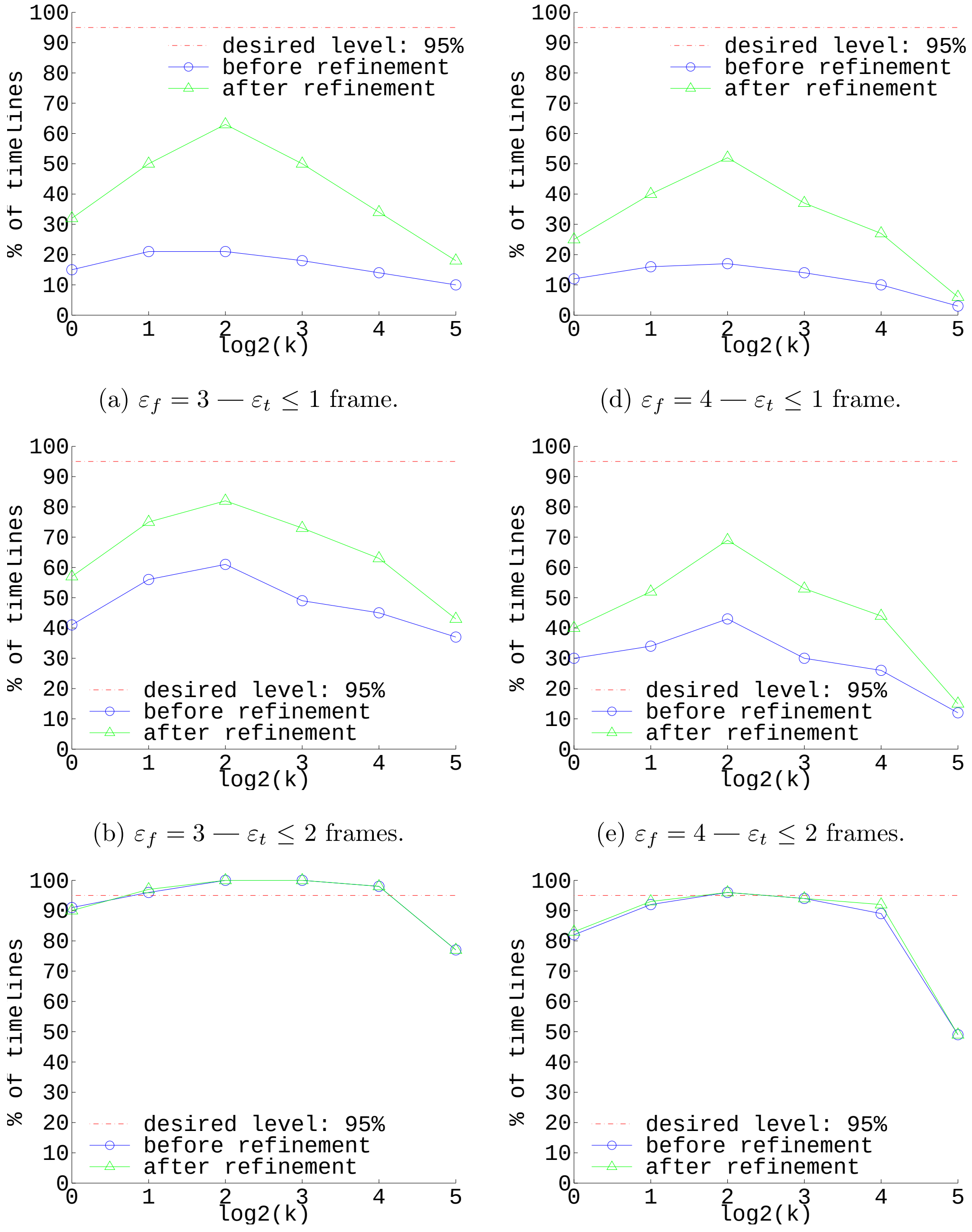
(b) $\varepsilon_f = 1$ — $\varepsilon_t \leq 2$ frames.

(e) $\varepsilon_f = 2$ — $\varepsilon_t \leq 2$ frames.

(c) $\varepsilon_f = 1$ — $\varepsilon_t \leq 5$ frames.

(f) $\varepsilon_f = 2$ — $\varepsilon_t \leq 5$ frames.

Figure 5.21: Percentages of timelines that lead to average temporal alignment errors smaller than or equal to 1, 2 and 5 frame(s), for $k = 1$, 2, 4, 8, 16 and 32 feature(s). (a), (b) and (c) Average error of the fundamental matrix: $\varepsilon_f = 1$ pixel. (d), (e) and (f) Average error of the fundamental matrix: $\varepsilon_f = 2$ pixels. In all cases, the tracker is corrupted by a gaussian noise with standard deviation of ±2 pixels ($R = 2$).



(a) $\varepsilon_f = 3$ — $\varepsilon_t \leq 1$ frame.

(d) $\varepsilon_f = 4$ — $\varepsilon_t \leq 1$ frame.

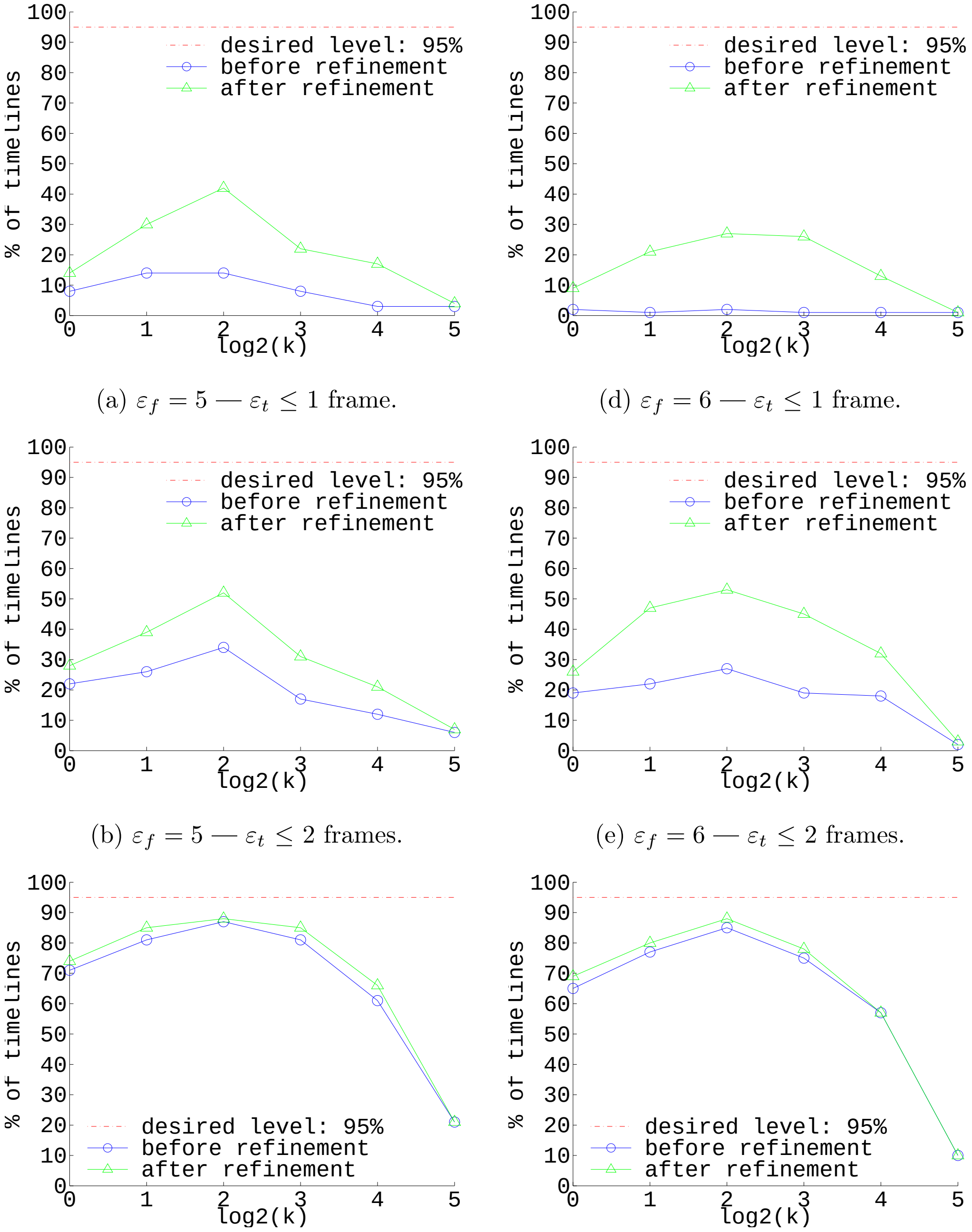
(b) $\varepsilon_f = 3$ — $\varepsilon_t \leq 2$ frames.

(e) $\varepsilon_f = 4$ — $\varepsilon_t \leq 2$ frames.

(c) $\varepsilon_f = 3$ — $\varepsilon_t \leq 5$ frames.

(f) $\varepsilon_f = 4$ — $\varepsilon_t \leq 5$ frames.

Figure 5.22: Percentages of timelines that lead to average temporal alignment errors smaller than or equal to 1, 2 and 5 frame(s), for $k = 1, 2, 4, 8, 16$ and 32 feature(s). (a), (b) and (c) Average error of the fundamental matrix: $\varepsilon_f = 3$ pixels. (d), (e) and (f) Average error of the fundamental matrix: $\varepsilon_f = 4$ pixels. In all cases, the tracker is corrupted by a gaussian noise with standard deviation of ±2 pixels ($R = 2$).



(a) $\varepsilon_f = 5$ — $\varepsilon_t \leq 1$ frame.

(d) $\varepsilon_f = 6$ — $\varepsilon_t \leq 1$ frame.

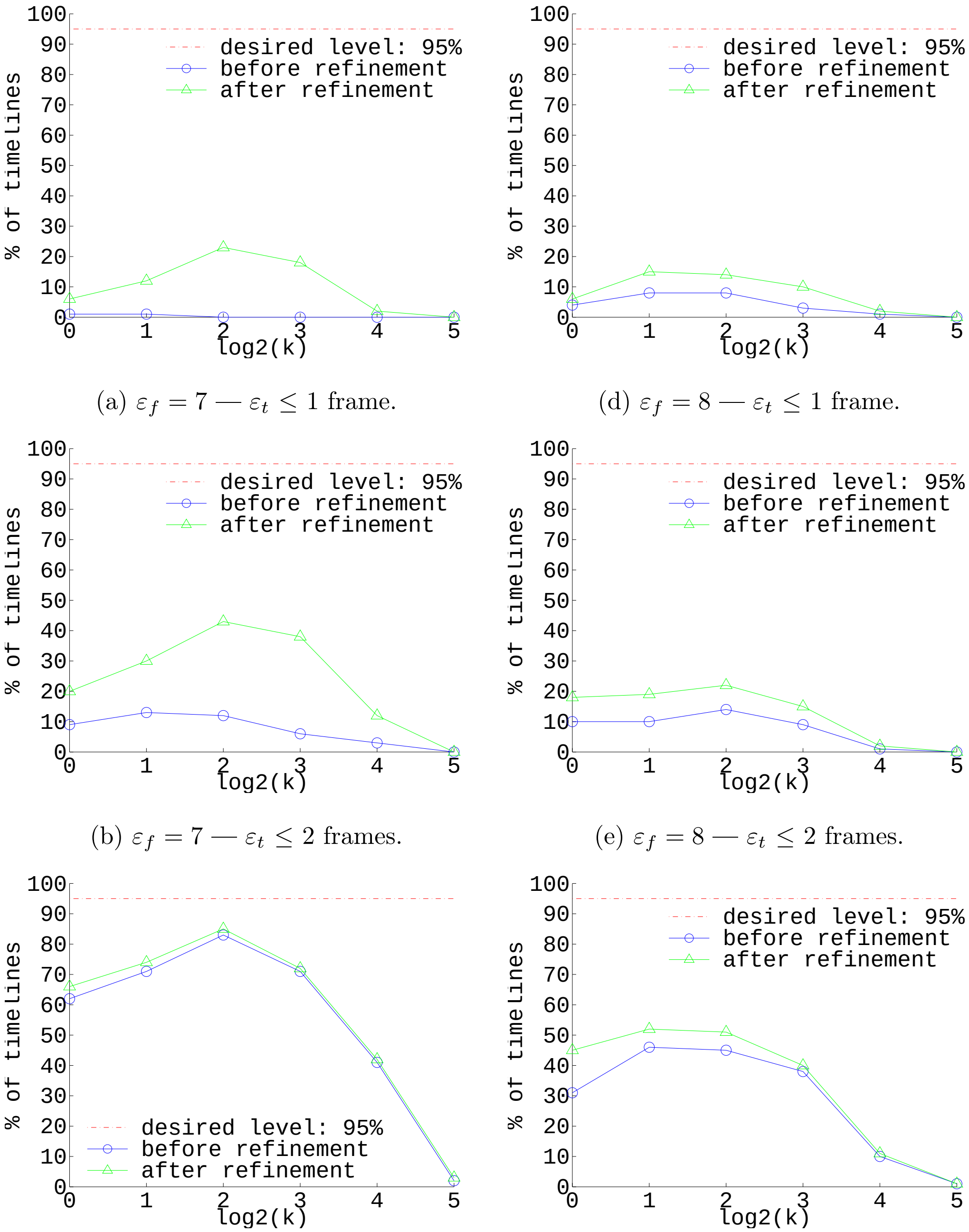
(b) $\varepsilon_f = 5$ — $\varepsilon_t \leq 2$ frames.

(e) $\varepsilon_f = 6$ — $\varepsilon_t \leq 2$ frames.

(c) $\varepsilon_f = 5$ — $\varepsilon_t \leq 5$ frames.

(f) $\varepsilon_f = 6$ — $\varepsilon_t \leq 5$ frames.

Figure 5.23: Percentages of timelines that lead to average temporal alignment errors smaller than or equal to 1, 2 and 5 frame(s), for $k = 1$, 2, 4, 8, 16 and 32 feature(s). (a), (b) and (c) Average error of the fundamental matrix: $\varepsilon_f = 5$ pixels. (d), (e) and (f) Average error of the fundamental matrix: $\varepsilon_f = 6$ pixels. In all cases, the tracker is corrupted by a gaussian noise with standard deviation of ±2 pixels ($R = 2$).



(a) $\varepsilon_f = 7$ — $\varepsilon_t \leq 1$ frame.

(d) $\varepsilon_f = 8$ — $\varepsilon_t \leq 1$ frame.

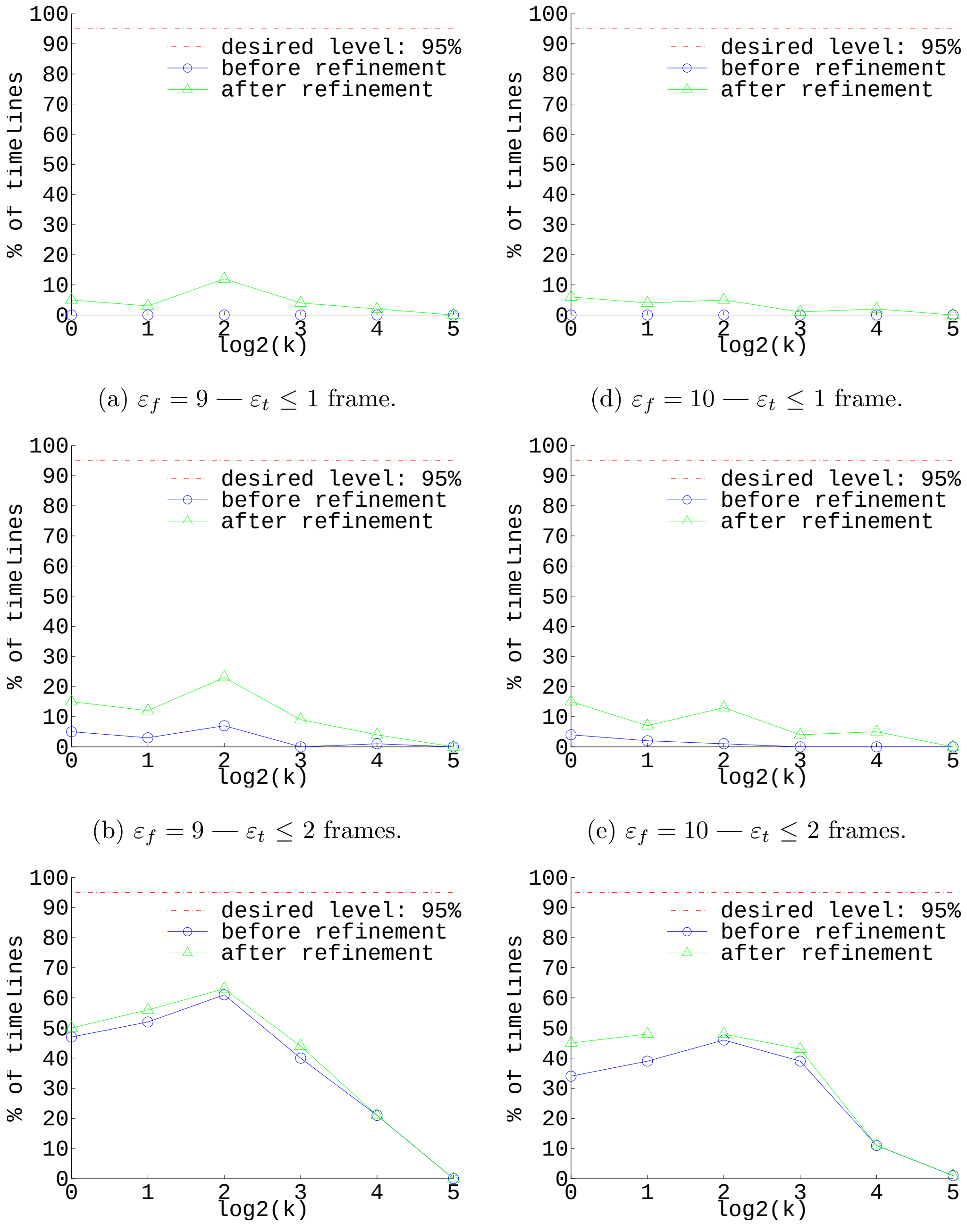
(b) $\varepsilon_f = 7$ — $\varepsilon_t \leq 2$ frames.

(e) $\varepsilon_f = 8$ — $\varepsilon_t \leq 2$ frames.

(c) $\varepsilon_f = 7$ — $\varepsilon_t \leq 5$ frames.

(f) $\varepsilon_f = 8$ — $\varepsilon_t \leq 5$ frames.

Figure 5.24: Percentages of timelines that lead to average temporal alignment errors smaller than or equal to 1, 2 and 5 frame(s), for k = 1, 2, 4, 8, 16 and 32 feature(s). (a), (b) and (c) Average error of the fundamental matrix: ε_f = 7 pixels. (d), (e) and (f) Average error of the fundamental matrix: $\varepsilon_f = 8$ pixels. In all cases, the tracker is corrupted by a gaussian noise with standard deviation of ±2 pixels ($R = 2$).



(a) $\varepsilon_f = 9$ — $\varepsilon_t \leq 1$ frame.

(d) $\varepsilon_f = 10$ — $\varepsilon_t \leq 1$ frame.

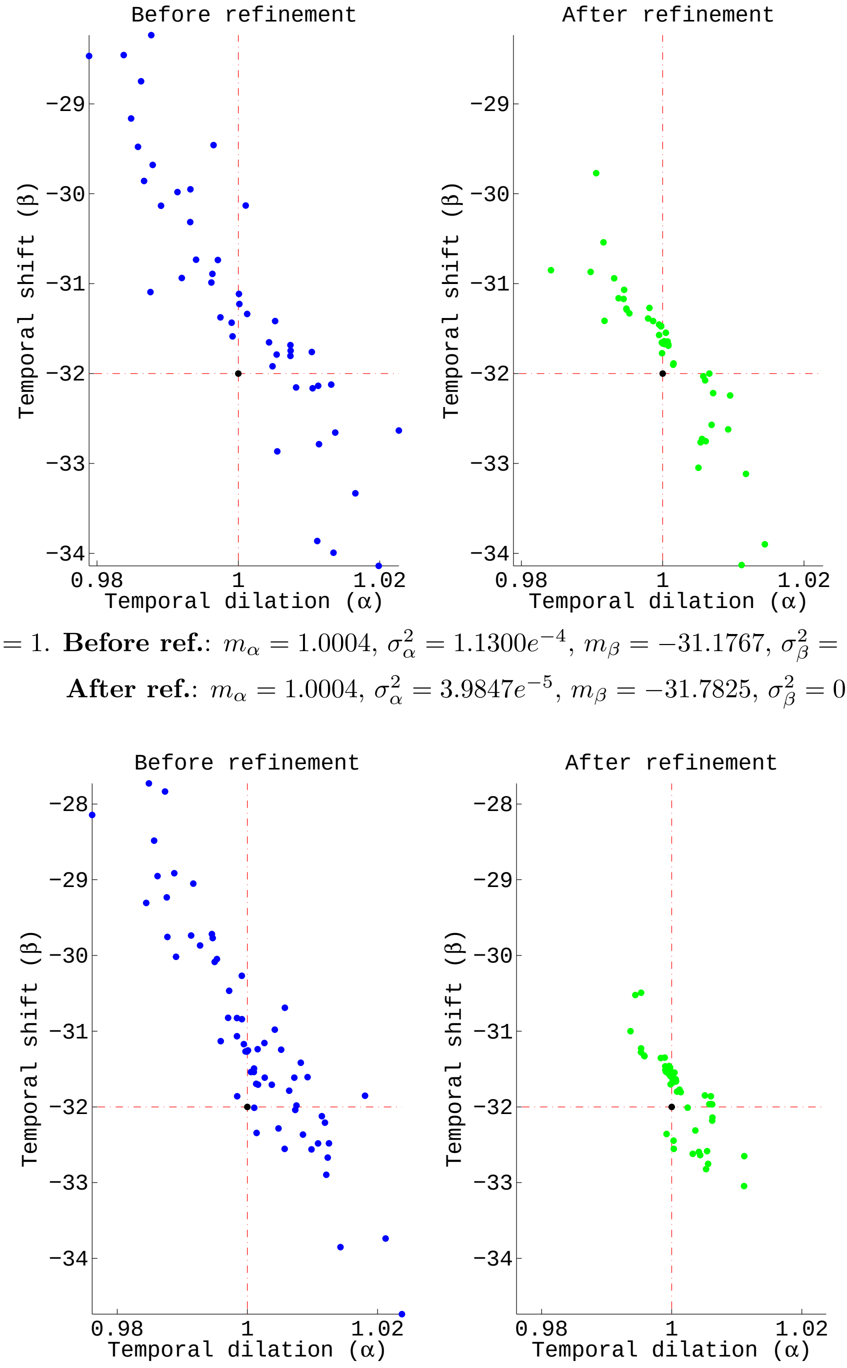
(b) $\varepsilon_f = 9$ — $\varepsilon_t \leq 2$ frames.

(e) $\varepsilon_f = 10$ — $\varepsilon_t \leq 2$ frames.

(c) $\varepsilon_f = 9$ — $\varepsilon_t \leq 5$ frames.

(f) $\varepsilon_f = 10$ — $\varepsilon_t \leq 5$ frames.

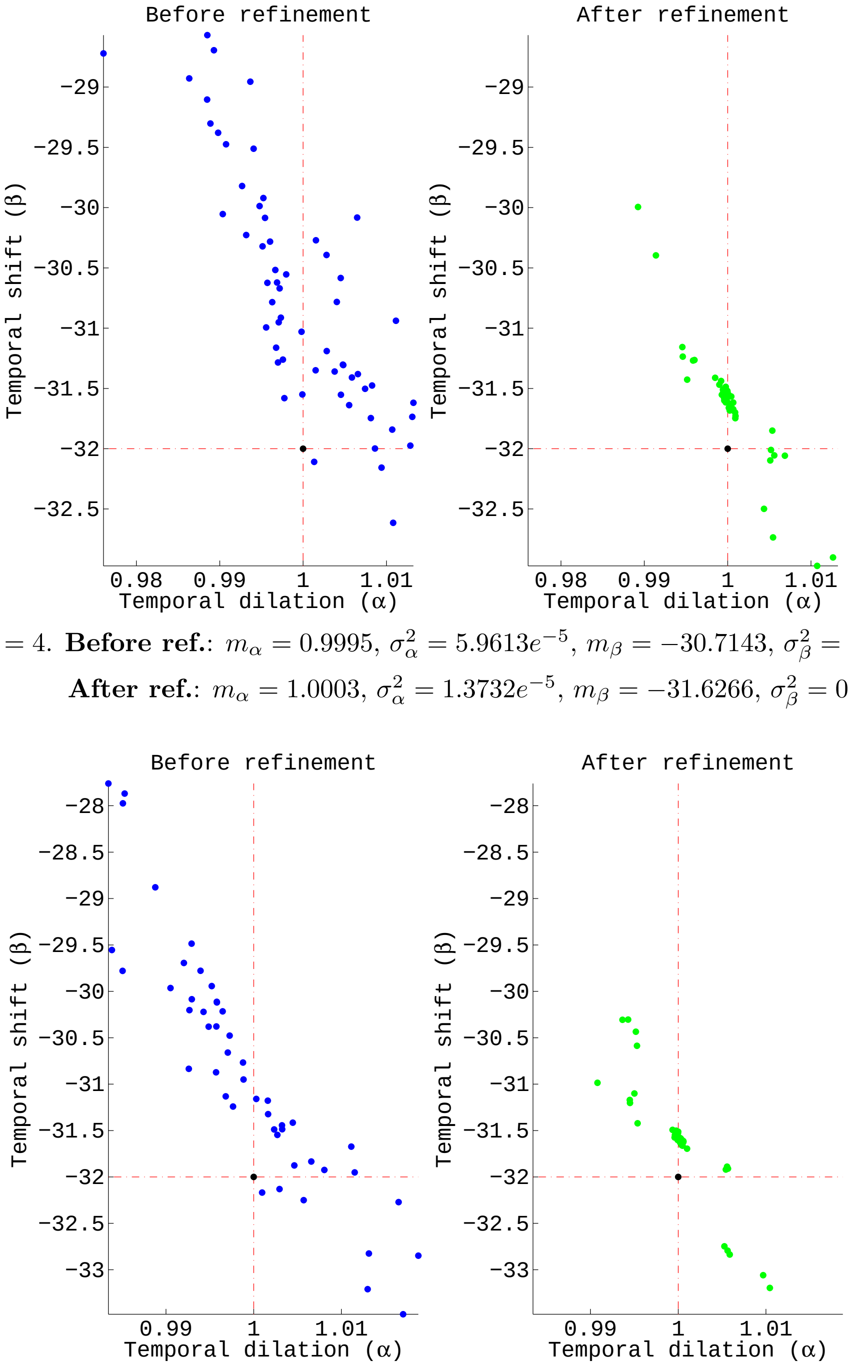
Figure 5.25: Percentages of timelines that lead to average temporal alignment errors smaller than or equal to 1, 2 and 5 frame(s), for $k = 1$, 2, 4, 8, 16 and 32 feature(s). (a), (b) and (c) Average error of the fundamental matrix: $\varepsilon_f = 9$ pixels. (d), (e) and (f) Average error of the fundamental matrix: $\varepsilon_f = 10$ pixels. In all cases, the tracker is corrupted by a gaussian noise with standard deviation of ±2 pixels ($R = 2$).



(a) $k = 1$. **Before ref.**: $m_\alpha = 1.0004$, $\sigma_\alpha^2 = 1.1300e^{-4}$, $m_\beta = -31.1767$, $\sigma_\beta^2 = 2.1003$.
After ref.: $m_\alpha = 1.0004$, $\sigma_\alpha^2 = 3.9847e^{-5}$, $m_\beta = -31.7825$, $\sigma_\beta^2 = 0.6514$.

(b) $k = 2$. **Before ref.**: $m_\alpha = 1.0008$, $\sigma_\alpha^2 = 8.9396e^{-5}$, $m_\beta = -31.0969$, $\sigma_\beta^2 = 2.0429$.
After ref.: $m_\alpha = 1.0009$, $\sigma_\alpha^2 = 1.2767e^{-5}$, $m_\beta = -31.7608$, $\sigma_\beta^2 = 0.2588$.

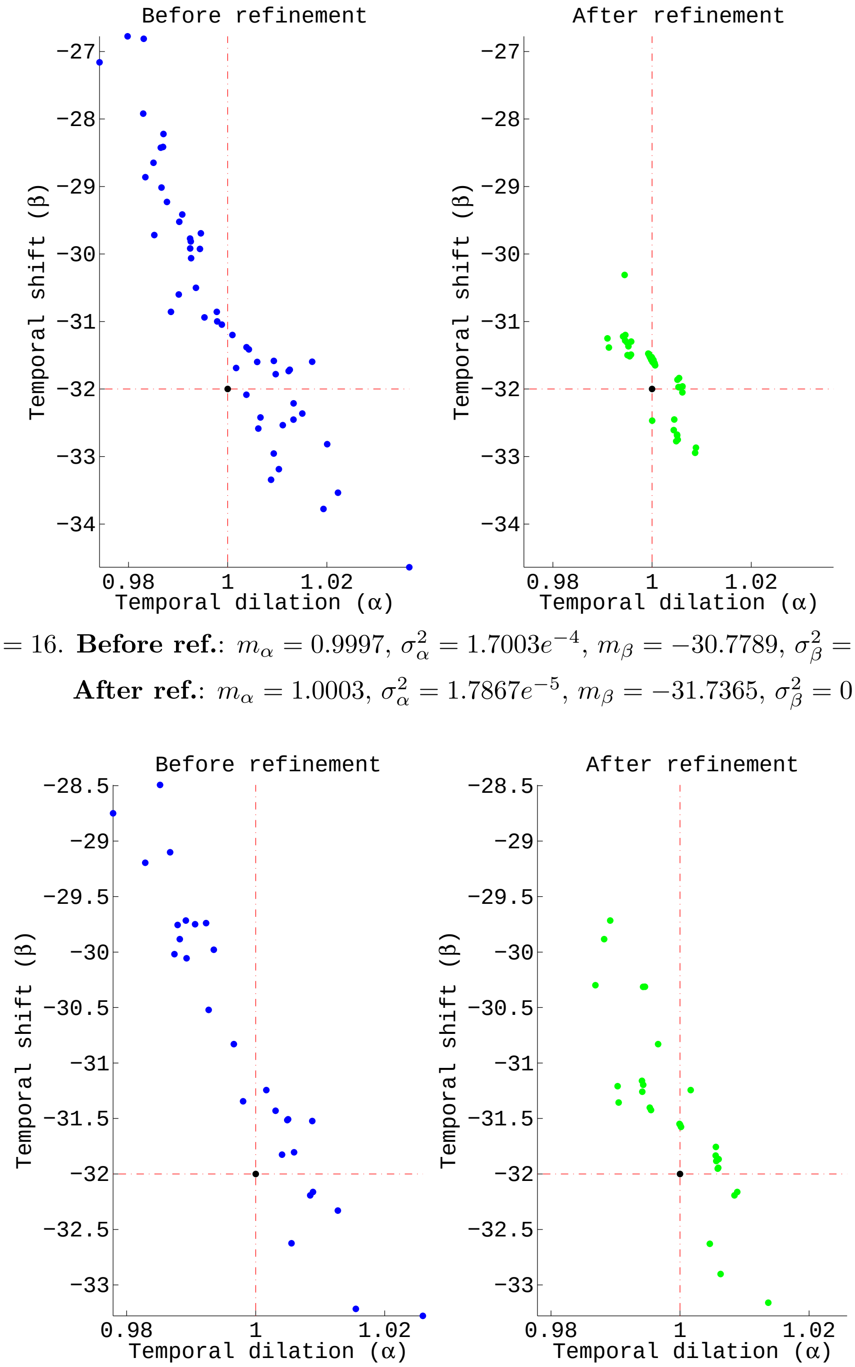
Figure 5.26: Space of estimated parameters for (a) $k = 1$ feature and (b) $k = 2$ features. Each point shown corresponds to the parameters of an estimated timeline with temporal alignment error smaller than or equal to 1 frame. m_α and m_β represent the average values and σ_α^2 and σ_β^2 are the variances of the temporal dilation (α) and the temporal shift (β), respectively. In both cases, the average error of the fundamental matrix and the standard deviation of the gaussian noise added to the tracker are 2 pixels.



(a) $k = 4$. **Before ref.**: $m_\alpha = 0.9995$, $\sigma_\alpha^2 = 5.9613e^{-5}$, $m_\beta = -30.7143$, $\sigma_\beta^2 = 0.9581$.
After ref.: $m_\alpha = 1.0003$, $\sigma_\alpha^2 = 1.3732e^{-5}$, $m_\beta = -31.6266$, $\sigma_\beta^2 = 0.1964$.

(b) $k = 8$. **Before ref.**: $m_\alpha = 0.9991$, $\sigma_\alpha^2 = 7.7272e^{-5}$, $m_\beta = -30.8503$, $\sigma_\beta^2 = 1.6661$.
After ref.: $m_\alpha = 1.0001$, $\sigma_\alpha^2 = 1.4973e^{-5}$, $m_\beta = -31.6023$, $\sigma_\beta^2 = 0.3413$.

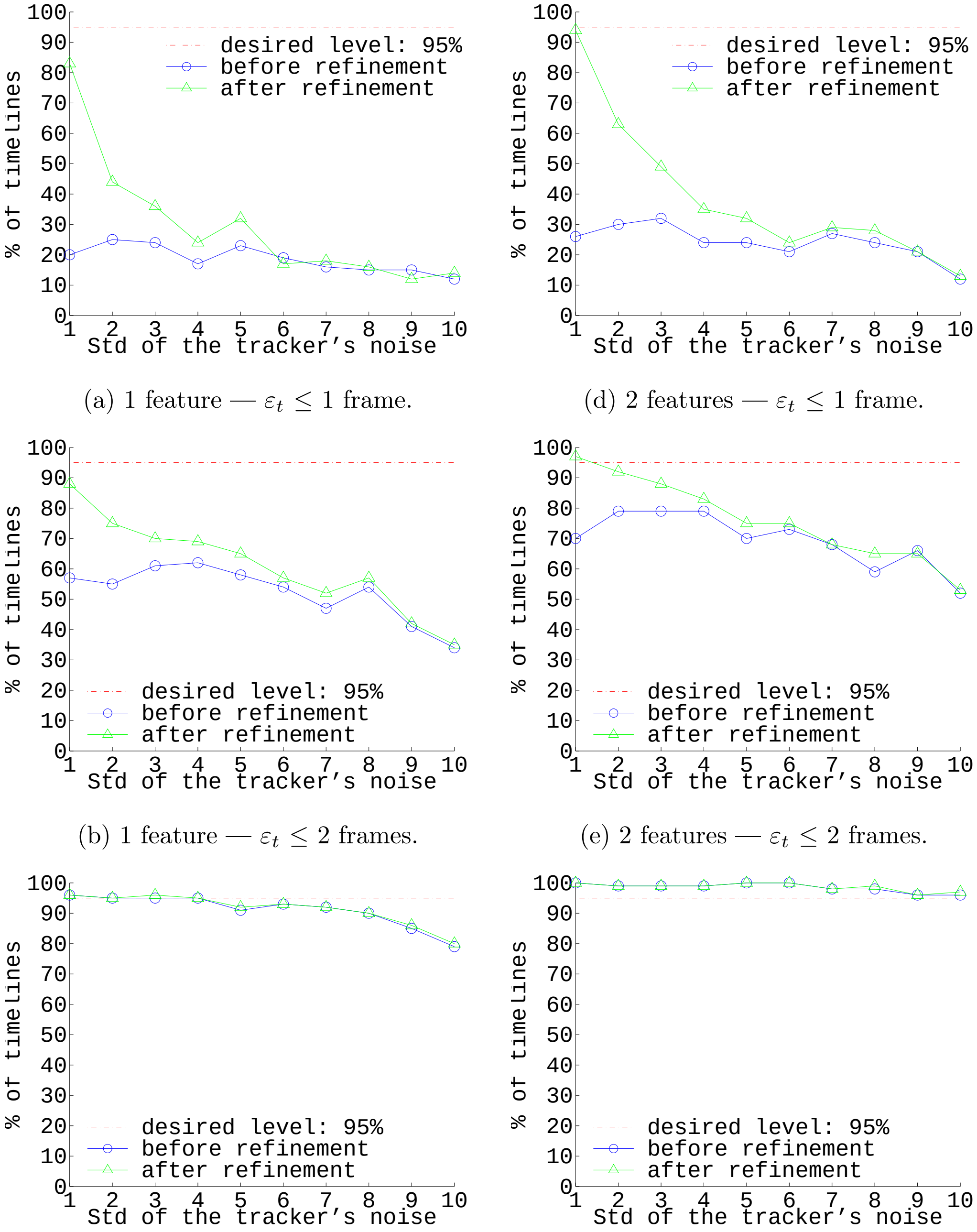
Figure 5.27: Space of estimated parameters for (a) $k = 4$ features and (b) $k = 8$ features. Each point shown corresponds to the parameters of an estimated timeline with temporal alignment error smaller than or equal to 1 frame. m_α and m_β represent the average values and σ_α^2 and σ_β^2 are the variances of the temporal dilation (α) and the temporal shift (β), respectively. In both cases, the average error of the fundamental matrix and the standard deviation of the gaussian noise added to the tracker are 2 pixels.



(a) $k = 16$. **Before ref.**: $m_\alpha = 0.9997$, $\sigma_\alpha^2 = 1.7003e^{-4}$, $m_\beta = -30.7789$, $\sigma_\beta^2 = 3.4365$.
After ref.: $m_\alpha = 1.0003$, $\sigma_\alpha^2 = 1.7867e^{-5}$, $m_\beta = -31.7365$, $\sigma_\beta^2 = 0.2663$.

(b) $k = 32$. **Before ref.**: $m_\alpha = 0.9982$, $\sigma_\alpha^2 = 1.2828e^{-4}$, $m_\beta = -30.8495$, $\sigma_\beta^2 = 1.7812$.
After ref.: $m_\alpha = 0.9994$, $\sigma_\alpha^2 = 5.2500e^{-5}$, $m_\beta = -31.4488$, $\sigma_\beta^2 = 0.6915$.

Figure 5.28: Space of estimated parameters for (a) $k = 16$ features and (b) $k = 32$ features. Each point shown corresponds to the parameters of an estimated timeline with temporal alignment error smaller than or equal to 1 frame. m_α and m_β represent the average values and σ_α^2 and σ_β^2 are the variances of the temporal dilation (α) and the temporal shift (β), respectively. In both cases, the average error of the fundamental matrix and the standard deviation of the gaussian noise added to the tracker are 2 pixels.



(a) 1 feature — $\varepsilon_t \leq 1$ frame.

(d) 2 features — $\varepsilon_t \leq 1$ frame.

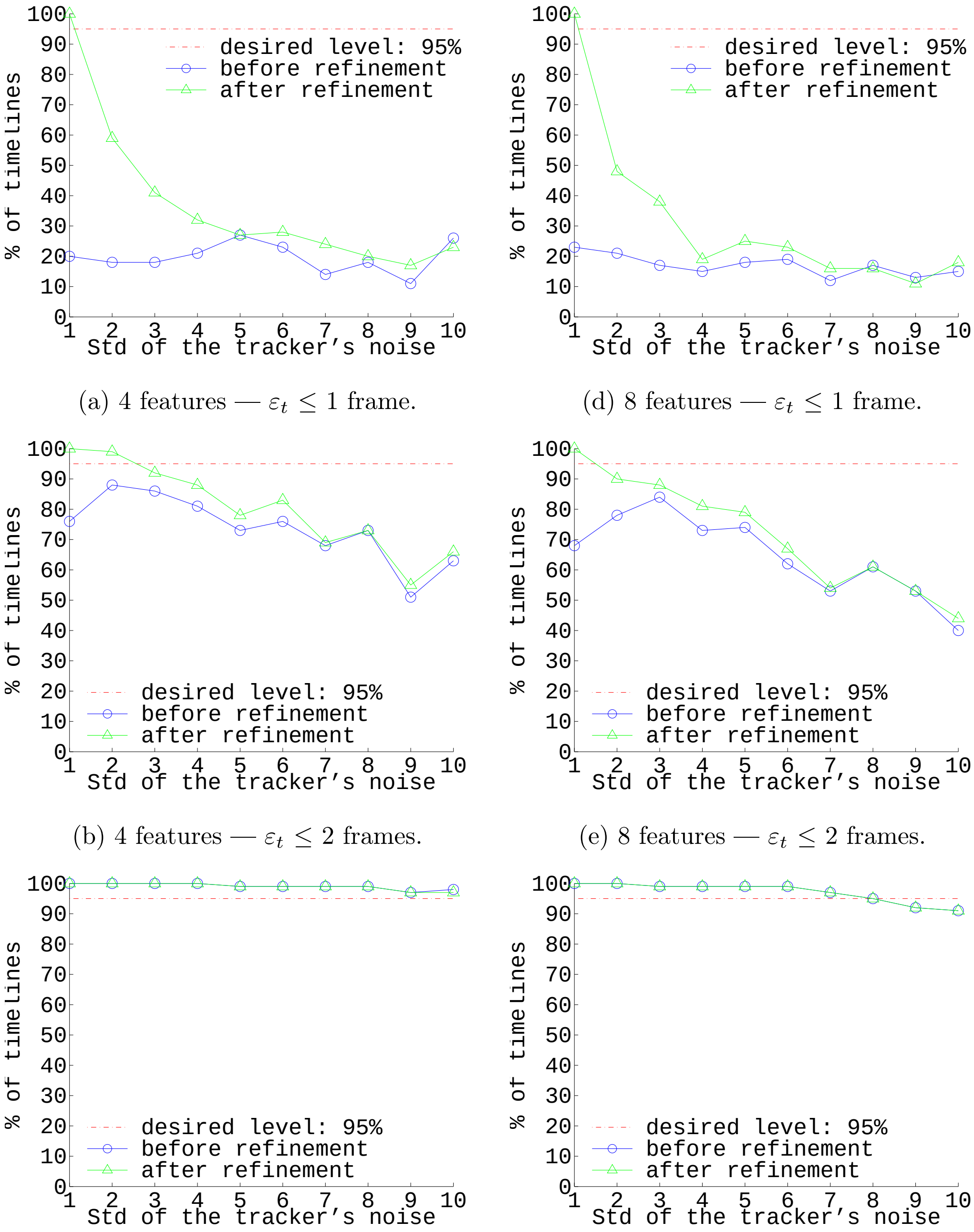
(b) 1 feature — $\varepsilon_t \leq 2$ frames.

(e) 2 features — $\varepsilon_t \leq 2$ frames.

(c) 1 feature — $\varepsilon_t \leq 5$ frames.

(f) 2 features — $\varepsilon_t \leq 5$ frames.

Figure 5.29: Percentages of timelines that lead to average temporal alignment errors smaller than or equal to 1, 2 and 5 frame(s), as a function of the tracker's noise level. (a), (b) and (c) Results for $k = 1$ feature. (d), (e) and (f) Results for $k = 2$ features. In all cases, a fundamental matrix with an average error of 2 pixels is used ($\varepsilon_f = 2$).



(a) 4 features — $\varepsilon_t \leq 1$ frame.

(d) 8 features — $\varepsilon_t \leq 1$ frame.

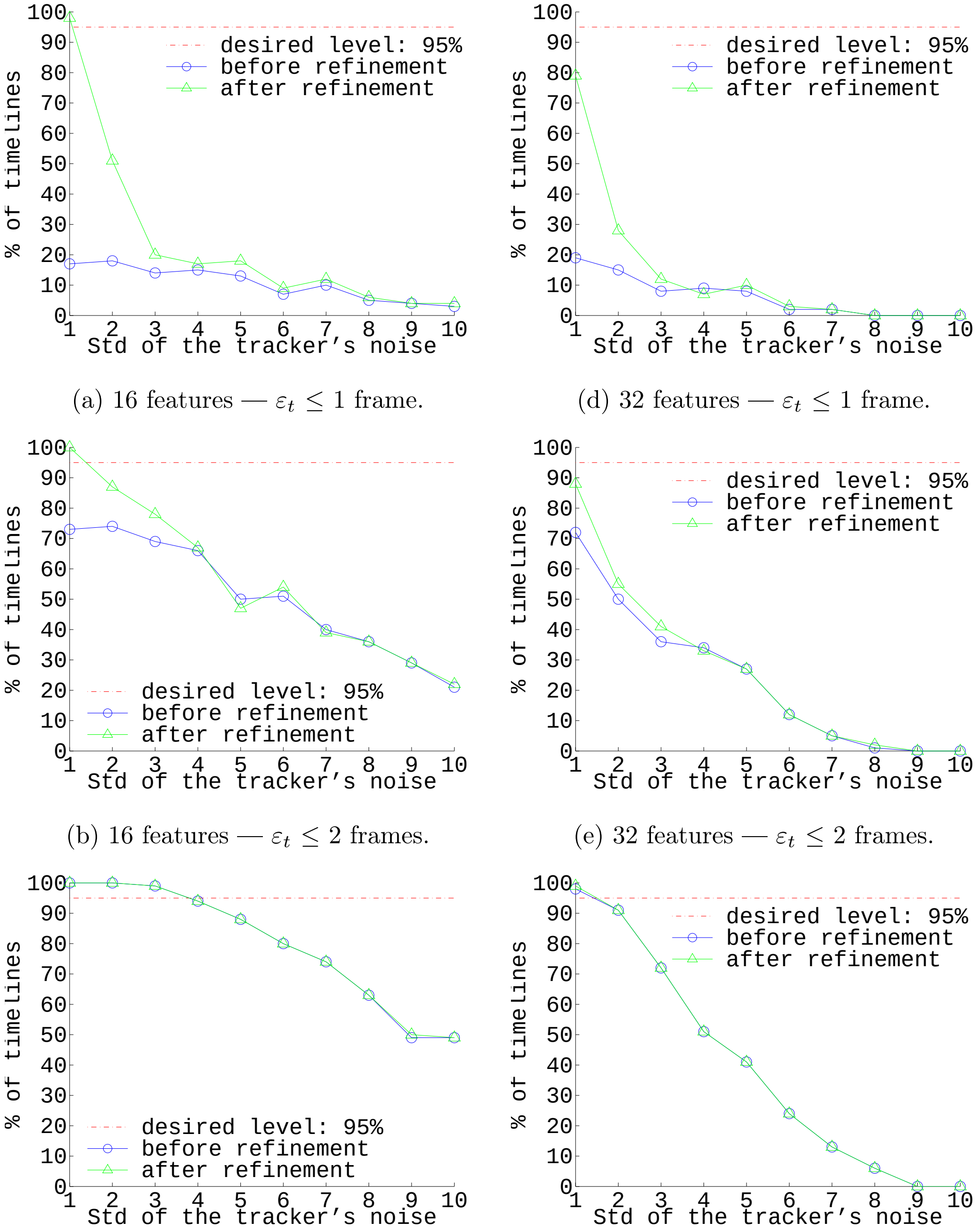
(b) 4 features — $\varepsilon_t \leq 2$ frames.

(e) 8 features — $\varepsilon_t \leq 2$ frames.

(c) 4 features — $\varepsilon_t \leq 5$ frames.

(f) 8 features — $\varepsilon_t \leq 5$ frames.

Figure 5.30: Percentages of timelines that lead to average temporal alignment errors smaller than or equal to 1, 2 and 5 frame(s), as a function of the tracker's noise level. (a), (b) and (c) Results for $k = 4$ features. (d), (e) and (f) Results for $k = 8$ features. In all cases, a fundamental matrix with an average error of 2 pixels is used ($\varepsilon_f = 2$).



(a) 16 features — $\varepsilon_t \leq 1$ frame.

(d) 32 features — $\varepsilon_t \leq 1$ frame.

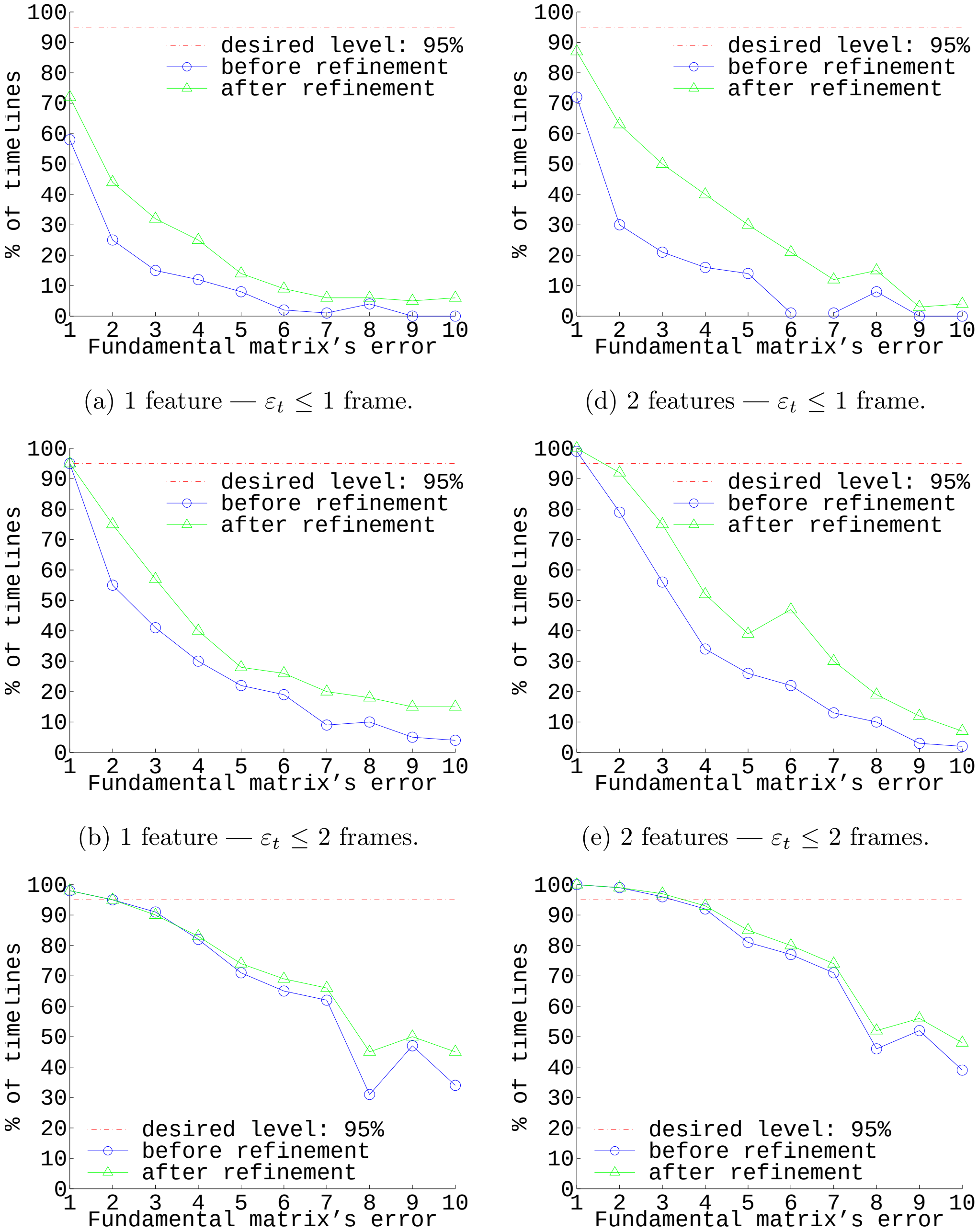
(b) 16 features — $\varepsilon_t \leq 2$ frames.

(e) 32 features — $\varepsilon_t \leq 2$ frames.

(c) 16 features — $\varepsilon_t \leq 5$ frames.

(f) 32 features — $\varepsilon_t \leq 5$ frames.

Figure 5.31: Percentages of timelines that lead to average temporal alignment errors smaller than or equal to 1, 2 and 5 frame(s), as a function of the tracker's noise level. (a), (b) and (c) Results for $k = 16$ features. (d), (e) and (f) Results for $k = 32$ features. In all cases, a fundamental matrix with an average error of 2 pixels is used ($\varepsilon_f = 2$).



(a) 1 feature — $\varepsilon_t \leq 1$ frame.

(d) 2 features — $\varepsilon_t \leq 1$ frame.

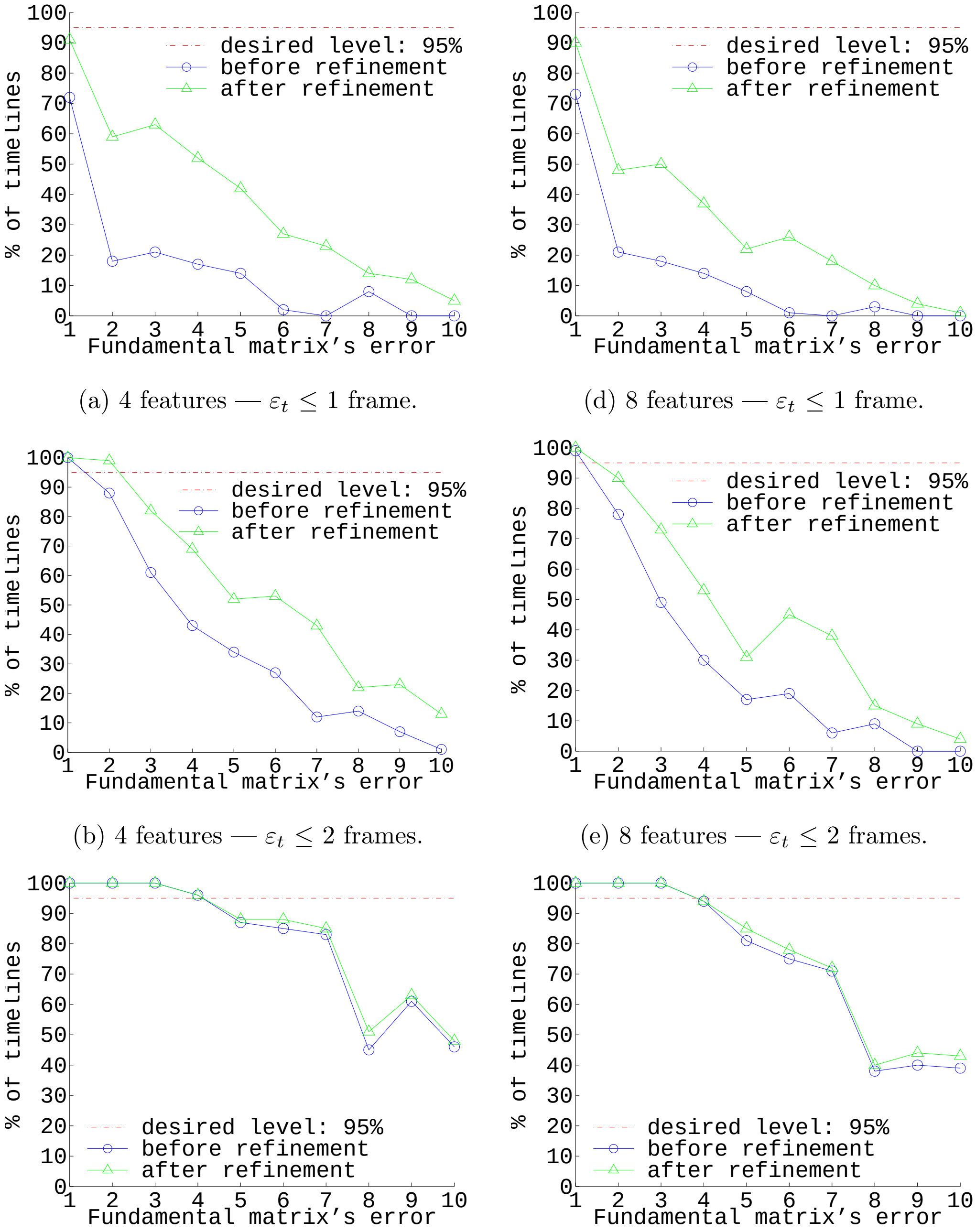
(b) 1 feature — $\varepsilon_t \leq 2$ frames.

(e) 2 features — $\varepsilon_t \leq 2$ frames.

(c) 1 feature — $\varepsilon_t \leq 5$ frames.

(f) 2 features — $\varepsilon_t \leq 5$ frames.

Figure 5.32: Percentages of timelines that lead to average temporal alignment errors smaller than or equal to 1, 2 and 5 frame(s), as a function of the average error of the fundamental matrix. (a), (b) and (c) Results for $k = 1$ feature. (d), (e) and (f) Results for $k = 2$ features. In all cases, the tracker is corrupted by a gaussian noise with standard deviation of ±2 pixels ($R = 2$).



(a) 4 features — $\varepsilon_t \leq 1$ frame.

(d) 8 features — $\varepsilon_t \leq 1$ frame.

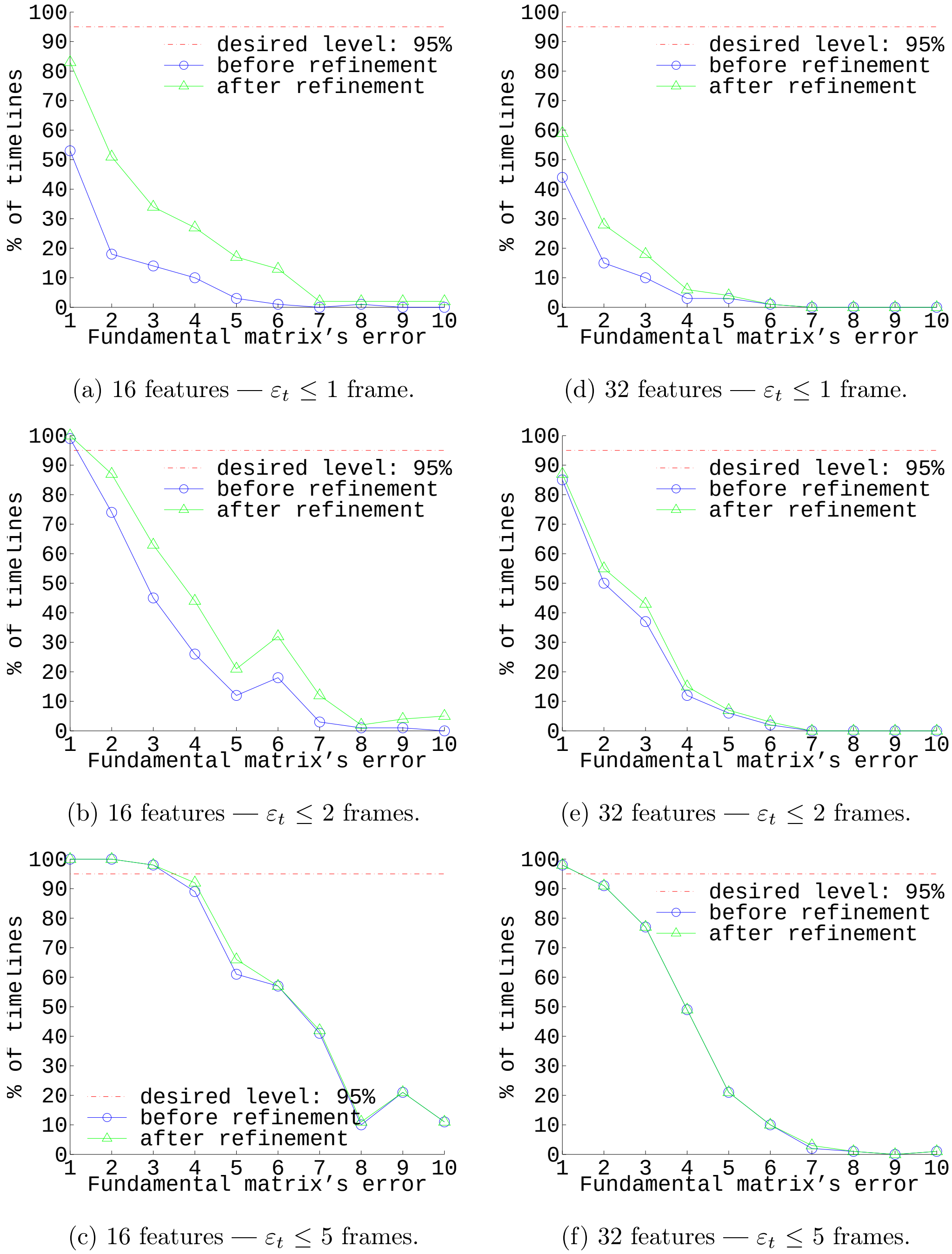
(b) 4 features — $\varepsilon_t \leq 2$ frames.

(e) 8 features — $\varepsilon_t \leq 2$ frames.

(c) 4 features — $\varepsilon_t \leq 5$ frames.

(f) 8 features — $\varepsilon_t \leq 5$ frames.

Figure 5.33: Percentages of timelines that lead to average temporal alignment errors smaller than or equal to 1, 2 and 5 frame(s), as a function of the average error of the fundamental matrix. (a), (b) and (c) Results for $k = 4$ features. (d), (e) and (f) Results for $k = 8$ features. In all cases, the tracker is corrupted by a gaussian noise with standard deviation of ± 2 pixels ($R = 2$).



(a) 16 features — $\varepsilon_t \leq 1$ frame.

(d) 32 features — $\varepsilon_t \leq 1$ frame.

(b) 16 features — $\varepsilon_t \leq 2$ frames.

(e) 32 features — $\varepsilon_t \leq 2$ frames.

(c) 16 features — $\varepsilon_t \leq 5$ frames.

(f) 32 features — $\varepsilon_t \leq 5$ frames.

Figure 5.34: Percentages of timelines that lead to average temporal alignment errors smaller than or equal to 1, 2 and 5 frame(s), as a function of the average error of the fundamental matrix. (a), (b) and (c) Results for $k = 16$ features. (d), (e) and (f) Results for $k = 32$ features. In all cases, the tracker is corrupted by a gaussian noise with standard deviation of ±2 pixels ($R = 2$).

Chapter 6

Conclusions

> Vivendo, se aprende, mas o que se aprende, mais, é só a fazer outras maiores perguntas.
>
> *João Guimarães Rosa*

6.1 Summary of the Accomplished Work

We have proposed a novel feature–based methodology for aligning both in time and space multiple video sequences acquired from distinct viewpoints. More specifically, a major contribution of our methodology is that it reduces the computation of temporal and spatio-temporal alignments between sequences to linear regression and linear optimization problems, while previous feature–based techniques need to search the entire space of possible temporal alignments. The quality of the computed alignments and the computational cost of our techniques are invariant to the magnitude of the initial temporal offsets between sequences. Moreover, unlike existing methods, which work for only two video sequences, our approach can handle an arbitrary number of sequences in a single step. Basically, our sequence–to–sequence alignment approach is constituted by two techniques: (a) one that builds large sets of temporal constraints from a rough spatial alignment between sequences and then performs a robust linear regression in the temporal domain to recover the globally correct temporal alignment, and (b) one that linearizes feature trajectories around the points of intersection with epipolar lines to reduce the problem of finding the complete spatio–temporal alignment between two sequences to a problem of solving a linear system.

We have shown that our work is suitable for solving several types of problems presented in current applications that benefit from the availability of simultaneous video recordings of the same physical event, proving that it is an interesting alternative, since that it is less expensive and easier to use outside labs than video synchronization hardware and it can be applied to various multi-view sequences that already exist in video databases, such as those of sport events.

From a theoretical standpoint, this work is relevant since it provided additional theoretical and empirical evidence that by considering temporal and spatial cues into a single alignment framework, many physical events which are inherently ambiguous for traditional image-to-image alignment methods, are uniquely resolved by sequence-to-sequence alignment techniques.

Our experimental results suggest that our timeline reconstruction algorithm provides a simple and effective method for temporally aligning multiple video sequences. Unlike previous approaches, it is able to handle temporal dilations and large time shifts, with no degradation in accuracy, even when scene points move along three-dimensional, overlapping and almost–cyclical trajectories. Importantly, by reducing the alignment problem to a RANSAC-based procedure, our algorithms are able to tolerate large proportions of outliers in the data, high levels of noise, discontinuities in feature trajectories, complete absence of stereo correspondences for moving features, and sequences that contain multiple frame rates. Finally, our approach requires the ability to track scene points only across two consecutive frames of the same sequence, what makes it robust to common tracking problems, such as occlusion, and it does not require the ability to establish feature correspondences between the sequences.

The above-mentioned contributions of the present work and the experiments performed with real-world sequences presented in Section 5.1 were published in one of the most important international peer reviewed conferences in computer vision:

- R. Carceroni, F. Pádua, G. Massahud, K. Kutulakos, "Linear Sequence-to-Sequence Alignment", in Proc. of IEEE Computer Vision and Pattern Recognition Conference, Washington, USA, pp. 746-753, 2004.

Moreover, we are currently preparing a complete journal article, which will be submitted to the *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*. This article includes a careful analysis about the scalability and accuracy of our approach, and discusses the experimental results obtained with synthetic sequences, which are described in Section 5.2.

6.2 Discussion

Although our experimental results demonstrate that we may apply our approach successfully in several challenging scenarios, it has some failure modes, which must be appointed. Firstly, our approach may not be applied in case of dynamic scenes where we cannot detect moving features (e.g., the monitored scene is an empty room whose light has been turned off), or in cases where the scene features move in such a way that their corresponding trajectories in the image plane are represented by a single point (e.g., a feature moves in a direction that is perpendicular to the image plane). In those scenarios, it would not be possible to generate the voting space for estimating the searched timeline. Another critical situation occurs when the epipolar geometry of a pair of cameras defines epipolar lines that do not intersect the feature trajectories (e.g., the epipolar lines are parallel to the trajectories). Again, in that scenario, we may not compute the space with the candidate temporal alignments. Fortunately, the above mentioned situations do not represent a significative portion of the scenarios that we may find in the practice.

Regarding the accuracy of our methodology, we believe that some simple additional changes in our technique could lead to even better results. Firstly, we believe that by using specific object characteristics, such as its color and texture, we could avoid the insertion of spurious information in the voting spaces created by our temporal alignment algorithm. In particular, a candidate point should concatenate coordinates f_i and f_j for cameras i and j, respectively, only if the intersection points that defined them belong to blobs that present the same characteristic of interest (e.g., color).

Moreover, we believe that in cases where it is possible to provide the optimization process formulated by our methodology with the *a priori* information about the possible range in which the true temporal misalignment is, we could improve the accuracy of the refined temporal parameters estimated. In this case, by using that *a priori* information, the optimization algorithm would know when it would be diverging from the actual solution and correction actions could be appropriately taken.

6.3 Future Work

Additional theoretical investigations need to be considered for future work. Firstly, the methodology proposed in Chapter 3 assumes that all cameras acquire frames at constant (albeit not necessarily identical) temporal sampling rates. Based on that assumption, our approach model the temporal misalignment between a pair of video sequences as an one-dimensional affine transformation. The pairwise temporal relations modelled by that transformation induce a global relationship between the frame numbers of the input sequences, which is captured by the N-dimensional line that we call timeline. However, such a kind of mathematical modelling is not appropriate when some sequences work with variable frame rates. Therefore, the development of an alternative mathematical model, which can couple with this problem represents an important topic for future research.

Also, it is necessary to conceive alternative techniques for obtaining initial estimates of the cameras' epipolar geometry. Currently, our approach is based on the use of background features whose image coordinates are processed by the normalized eight-point algorithm in order to estimate the fundamental matrices that capture the geometric relations between the views. However, there are some cases where we can not identify enough static scene points for every pair of video sequences, making impossible the computation of an initial estimate of the cameras' epipolar geometry and, consequently, the use of our methodology.

Finally, another important direction for future work is 3D scene reconstruction. By combining our temporal alignment approach with multi-view stereo techniques, important advances could be achieved in the development of robust systems for reconstructing 3D dynamic scenes, specially the ones presented in old video footage, where multiple replays of the same event are shown from different viewpoints. Moreover, as the capabilities of video standards and receiver hardware are increasing towards integrated 3D animations, generating realistic content is now becoming a limiting factor. In this context, an increasing demand has been verified on techniques for generating 3D content from reality, i.e., from video sequences acquired with TV cameras, providing the TV viewer with animated 3D reconstructions of physical events and allowing for an immersive experience via free interaction on the receiver side.

Bibliography

Atkinson, K. (1989). *An Introduction to Numerical Analysis.* John Wiley and Sons, second edition.

Brown, M. and Lowe, D. (2003). Recognising Panoramas. In *Proc. IEEE International Conference on Computer Vision*, pages 1218–1225.

Canterakis, N. (2000). A Minimal Set of Constraints for the Trifocal Tensor. In *Proc. European Conference on Computer Vision*, pages 84–99. Springer LNCS 1842.

Cantzler, H. (2004). Random Sample Consensus (RANSAC). Technical report, Institute for Perception, Action and Behaviour, Division of Informatics, University of Edinburgh.

Carceroni, R. L. (2001). *Recovering Non-Rigid 3D Motion, Shape and Reflectance from Multi-View Image Sequences: A Differential-Geometric Approach.* PhD thesis, Department of Computer Science - The College Arts and Sciences, University of Rochester, Rochester, New York.

Caspi, Y. and Irani, M. (2000). A Step Towards Sequence-to-Sequence Alignment. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 682–689, Hilton Head Island, South Carolina.

Caspi, Y. and Irani, M. (2001). Alignment of Non-Overlapping Sequences. In *Proc. IEEE International Conference on Computer Vision*, volume 2, pages 76–83, Vancouver, Canada.

Caspi, Y., Simakov, D., and Irani, M. (2002). Feature-Based Sequence-to-Sequence Matching. In *VAMODS (Vision and Modelling of Dynamic Scenes) workshop with ECCV*, Copenhagen, Denmark.

Clarke, J., Carlsson, S., and Zisserman, A. (1996). Detecting and Tracking Linear Features Efficiently. In *Proc. British Machine Vision Conference*, pages 415–424.

Faugeras, O. and Mourrain, B. (1995). On the Geometry and Algebra of the Point and Line Correspondences Between N Images. In *Proc. IEEE International Conference on Computer Vision*, pages 951–956.

Faugeras, O. and Papadopoulo, T. (1997). Grassmann-Cayley Algebra for Modeling Systems of Cameras and the Algebraic Equations of the Manifold of Trifocal Tensors. Technical Report 3225, INRIA, Sophia-Antipolis, France.

FIFA (2002). FIFA World Cup Archives: Goal of the Century. `http://fifaworldcup.yahoo.com/02/en/pf/h/gotc/launch.html`.

FIFA (2004a). FIFA World Cup Archives: Classic Games. `http://fifaworldcup.yahoo.com/06/en/p/cg/index.html`.

FIFA (2004b). FIFA World Cup Archives: Photo Gallery. `http://fifaworldcup.yahoo.com/02/en/011221/4/44c.html`.

Fischler, M. and Bolles, R. (1981). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395.

Forsyth, D. and Ponce, J. (2002). *Computer Vision: A Modern Approach.* Prentice Hall, ISBN: 0130851981, first edition.

Hartley, R. (1997a). In Defense of the Eight-point Algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593.

Hartley, R. (1997b). Lines and Points in Three Views and the Trifocal Tensor. *International Journal of Computer Vision*, 22(2):125–140.

Hartley, R. (1998). Computation of the Quadrifocal Tensor. In *Proc. European Conference on Computer Vision*, pages 20–35. Springer-Verlag.

Hartley, R. and Zisserman, A. (2003). *Multiple View Geometry in Computer Vision.* Cambridge University Press, ISBN: 0521540518, second edition.

Hefferon, J. (2001). *Linear Algebra.* Mathematics Department of Saint Michael's College, first edition.

Heyden, A. (1995a). *Geometry and Algebra of Multiple Projective Transformations.* PhD thesis, Department of Mathematics, Lund University, weden.

Heyden, A. (1995b). Reconstruction from Multiple Images Using Kinetic Depths. *International Journal of Computer Vision.*

Heyden, A. (1998). Algebraic Varieties in Multiple View Geometry. In *Proc. 5th European Conference on Computer Vision*, pages 3–19, Freiburg, Germany.

Heyden, A. and Åström, K. (1997). Algebraic Properties of Multilinear Constraints. *Mathematical Methods in the Applied Sciences*, 20:1135–1162.

Horst, J. and Beichl, I. (1997). A Simple Algorithm for Efficient Piecewise Linear Approximation of Space Curves. In *Proc. IEEE International Conference on Image Processing.*

Isard, M. and MacCormick, J. (2001). BraMBLe: A Bayesian Multiple-Blob Tracker. In *Proc. IEEE International Conference on Computer Vision*, volume 2, pages 34–41, Vancouver, Canada.

Jepson, A., Fleet, D., and El-Maraghi, T. (2003). Robust On-line Appearance Models for Visual Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1296–1311.

Kutulakos, K. and Seitz, S. (2000). A Theory of Shape by Space Carving. *International Journal of Computer Vision*, 38(3):199–218.

Kutulakos, K. N. (2000). Approximate N-View Stereo. In *Proc. European Conference on Computer Vision*, pages 67–83.

Lee, L., Romano, R., and Stein, G. (2000). Monitoring Activities from Multiple Video Streams: Establishing a Common Coordinate Frame. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22:758–767.

Longuet-Higgins, H. (1981). A Computer Algorithm for Reconstructing a Scene from Two Projections. *Nature*, 293:133–135.

Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110.

Luong, Q.-T. and Vieville, T. (1994). Canonic Representations for the Geometries of Multiple Projective Views. In *Proc. 4th European Conference on Computer Vision*, volume 800, pages 589–599, Cambridge, UK.

Ma, Y., Soatto, S., Kosecka, J., and Sastry, S. S. (2003). *An Invitation to 3-D Vision - From Images to Geometric Models.* Springer-Verlag, first edition.

Matas, J. and Chum, O. (2004). Randomized RANSAC with $t_{d,d}$ Test. *Image and Vision Computing*, 22(10):837–842.

McLauchlan, P. and Jaenicke, A. (2002). Image Mosaicing Using Sequential Bundle Adjustment. *Image and Vision Computing*, 20(9-10):751–759.

Myatt, D., Torr, P., Nasuto, S., Bishop, J., and Craddock, R. (2002). Napsac: High Noise, High Dimensional Robust Estimation - It's in the Bag. In *Proc. British Machine Vision Conference*, pages 458–467.

Press, W., Flannery, B., Teukolsky, S., and Vetterling, W. (1988). *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, first edition.

Pritchett, P. and Zisserman, A. (1998). Wide Baseline Stereo Matching. In *Proc. IEEE International Conference on Computer Vision*, pages 754–760.

Rao, C., Gritai, A., Shah, M., and Syeda-Mahmood, T. (2003). View-Invariant Alignment and Matching of Video Sequences. In *Proc. of IEEE International Conference on Computer Vision*, volume 2, pages 939–945, Nice,France.

Reid, I. and Zisserman, A. (1996). Goal Directed Video Metrology. In *Proc. European Conference on Computer Vision*, pages 647–658.

Schaffalitzky, F. and Zisserman, A. (2001). Viewpoint Invariant Texture Matching and Wide Baseline Stereo. In *Proc. 8th IEEE International Conference on Computer Vision*, Vancouver, Canada.

Sharipov, R. (2004). *Quick Introduction to Tensor Analysis.* Freely distributed on-line - http://samizdat.mines.edu/tensors/, first edition.

Shashua, A. and Werman, M. (1995). Fundamental Tensor: On the Geometry of Three Perspective Views. In *Proc. IEEE International Conference on Computer Vision*, pages 920–925.

Shechtman, E., Caspi, Y., and Irani, M. (2002). Increasing Video Resolution in Time and Space. In *Proc. European Conference in Computer Vision*, Copenhagen, Denmark.

Shi, J. and Tomasi, C. (1994). Good Features to Track. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Seattle.

Stein, G. (1998). Tracking from Multiple View Points: Self-Calibration of Space and Time. In *DARPA Image Understanding Workshop*, pages 521–527, Montery, Canada.

Szeliski, R. (1999). A Multi-View Approach to Motion and Stereo. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 157–163.

Tomasi, C. (2000). Mathematical Methods for Robotics and Vision. Technical Report CS 205, Stanford University.

Torr, P. (1995). *Outlier Detection and Motion Segmentation.* PhD thesis, Department of Engineering Science, University of Oxford.

Torr, P. and Zisserman, A. (1997). Robust Parametrization and Computation of the Trifocal Tensor. *Image and Vision Computing*, 15:591–605.

Torr, P. and Zisserman, A. (1999). Feature-Based Methods for Structure and Motion Estimation. In *International Workshop on Vision Algorithms*, pages 278–295.

Tresadern, P. and Reid, I. (2003). Synchronizing Image Sequences of Non-rigid Objects. In *Proc. British Machine Vision Conference*, volume 2, pages 629–638, Norwich.

Triggs, B. (1995). Matching Constraints and the Joint Image. In *Proc. IEEE International Conference on Computer Vision*, pages 338–343.

Trucco, E. and Verri, A. (1998). *Introductory Techniques for 3-D Computer Vision.* Prentice Hall, ISBN: 0132611082, first edition.

Vedula, S., Baker, S., and Kanade, T. (2002). Spatio-Temporal View Interpolation. In *Proc. Eurographics Workshop on Rendering*, pages 65–76.

Viéville, T. and Luong, Q. (1993). Motion of Points and Lines in the Uncalibrated Case. Technical Report RR-2054, INRIA.

Wolf, L. and Zomet, A. (2002a). Correspondence-Free Synchronization and Reconstruction in a Non-Rigid Scene. In *Workshop on Vision and Modelling of Dynamic Scenes*, Copenhagen, Denmark.

Wolf, L. and Zomet, A. (2002b). Sequence-to-Sequence Self Calibration. In *Proc. European Conference on Computer Vision*, volume 2, pages 370–382.

Zelnik-Manor, L. and Irani, M. (2001). Event-Based Analysis of Video. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Kauai,Hawaii USA.

Zhang, Z. (1998). Determining the Epipolar Geometry and its Uncertainty - A Review. *International Journal of Computer Vision*, 27(2):161–195.

Appendix A

The RANSAC Algorithm

The Random Sample Consensus (RANSAC) algorithm is a paradigm for robust fitting of models that was introduced by Fischler and Bolles in 1981 (Fischler and Bolles, 1981). It is robust in the sense of good tolerance to outliers in the experimental data. It is capable of interpreting and smoothing data containing a significant percentage of gross errors (Fischler and Bolles, 1981). The estimate is only correct with a certain probability, since RANSAC is a randomised estimator. The algorithm has been applied to a wide range of model parameters estimation problems in computer vision, such as detection of geometric primitives (Clarke et al., 1996), mosaicing (McLauchlan and Jaenicke, 2002), wide baseline stereo matching (Schaffalitzky and Zisserman, 2001; Pritchett and Zisserman, 1998) and motion segmentation (Torr, 1995).

Although RANSAC is a quite simple algorithm, it is a very powerful tool. In an iterative way, it randomly selects subsets from the input data and compute the model parameters that best fit to the sample (Matas and Chum, 2004). Samples are drawn uniformly from the input data set. Each point has the same probability of selection (uniform point sampling). For each sample a model hypothesis is constructed by computing the model parameters using the sample data. The size of the sample depends on the model one wants to find. Typically, it is the smallest size sufficient for determining model parameters (Matas and Chum, 2004). For example, to find circles in the data set, one has to draw three points, since three points are required to determine the parameters of a circle.

A.1 Hypotheses evaluation

In a second phase, the quality of the model parameters is evaluated on the full data set. Different cost functions may be used for the evaluation, the standard being the number of data points consistent with the model (Matas and Chum, 2004). The process is terminated when the likelihood of finding a better model becomes low (Fischler and Bolles, 1981). Usually, the model parameters estimated by RANSAC are not very precise. Therefore, the estimated model parameters are recomputed by, for example, a least-squares fit to the data subset which supports the best estimate. The input data may support several distinct models. In this case, the model parameters for the first model are estimated, the data points supporting the model are removed from the input data and the algorithm is simply repeated with the remainder of the data set to find the next best model. The strength of the algorithm is that it is likely to draw at least one set of points which consists only of inliers (Cantzler, 2004). Depending on the size of random samples, RANSAC can handle contamination levels well above 50%, which is commonly assumed to be a practical limit in robust statistics (Matas and Chum, 2004).

A.2 RANSAC parameters

The RANSAC technique uses three parameters to control the model estimation process (Cantzler, 2004). The first parameter (ϵ) is an error tolerance that determines a volume within which all compatible points must fall in. The second parameter (p) is the probability that at least one of our random selections is an error-free set of candidates. Finally, the third parameter (r) is the probability that a randomly-selected candidate is an inlier. The parameters p and r define the number of iterations z of RANSAC, as follows:

$$z = \left\lceil \frac{\log(1-p)}{\log(1-r^2)} \right\rceil , \tag{A.1}$$

Equation (A.1) expresses the fact that z should be large enough to ensure that, with probability p, at least one randomly-selected set of candidates is an inlier.

A.3 RANSAC efficiency

The speed of RANSAC depends on two factors (Matas and Chum, 2004). Firstly, the number of samples which have to be drawn to guarantee a certain confidence to obtain a good estimate, and secondly, the time spent evaluating the quality of each hypothetical model. The latter is proportional to the size of the data set.

Typically, a very large number of erroneous model parameters obtained from contaminated samples are evaluated. Such models are consistent with only a small fraction of the data (Cantzler, 2004). The evaluation of the models can be computationally optimised by randomising the evaluation (Matas and Chum, 2004). Every hypothetical model is first tested only with a small number of random data points from the data set. If a model does not get enough support from this random point set, then one can assume with a high confidence that the model is not a good estimate. Models passing the randomised evaluation are then evaluated on the full data set.

The performance of RANSAC degrades with increasing sample size or in case multiple models are supported by the data due to the decreasing probability of sampling a set that is composed entirely of inliers. A very common observation is that outliers possess a difuse distribution. On the other hand, inliers will tend to be located closely together. Therefore, the uniform sampling of points is replaced by selection of sample sets based on proximity taking spatial relationships into account (Myatt et al., 2002). The first initial sample point is selected randomly. The rest of the points are random points lying within a hypersphere centred on the first point. The selection of sample sets of adjacent points can significantly improve the probability of selecting a set of inliers and thus reduce the number of samples required to find a good model estimate (Matas and Chum, 2004).

Appendix B

Tensorial Notation

This brief introduction on tensor notation is based on the explanations presented in Hartley (1997b) and Hartley and Zisserman (2003). For more details about this topic, the reader is referred to Sharipov (2004) and Triggs (1995). For the sake of simplicity, we will present the concepts envolved in tensorial notation in the context of low-dimensional projective spaces, rather than in a general context, exactly as performed in Hartley (1997a) and Hartley and Zisserman (2003).

Consider a set of basis vectors $\mathbf{e}_i$, i=1,...,3 for a 2-dimensional projective space $\mathcal{P}^2$. Let their indices be written as subscripts. With respect to this basis, a point in $\mathcal{P}^2$ is represented by a set of coordinates q^i, which represents the point $\Sigma_{i=1}^3 q^i \mathbf{e}_i$. The coordinates are written with an upper index. Let $\mathbf{q}$ represent the triple of coordinates, $\mathbf{q} = (q^1, q^2, q^3)^\top$.

Now, consider a change of coordinate axes in which the basis vectors $\mathbf{e}_i$ are replaced by a new basis set $\hat{\mathbf{e}}_j$, where $\hat{\mathbf{e}}_j = \Sigma_i H_j^i \mathbf{e}_i$, and H is the basis transformation matrix with entries H_j^i. If $\hat{\mathbf{q}} = (\hat{q}^1, \hat{q}^2, \hat{q}^3)^\top$ are the coordinates of the vector with respect to the new basis, then we may verify that $\hat{\mathbf{q}} = H^{-1}\mathbf{q}$. Thus, if the basis vectors transform according to H the coordinates of points transform according to the inverse transform H^{-1}.

Next, consider a line in $\mathcal{P}^2$ represented by coordinates $\boldsymbol{\lambda}$ with respect to the original basis. With respect to new basis, it may be verified that the line is respresented by a new set of coordinates $\hat{\boldsymbol{\lambda}} = H^\top \boldsymbol{\lambda}$. Thus coordinates of the line transform according to $H^\top$.

Finally, let $\mathcal{P}$ be a matrix representing a mapping between vector spaces. If G and H represent basis transformations in the domain and range spaces, then with respect to the new bases, the mapping is represented by a new matrix $\hat{P} = H^{-1}PG$. Note in these examples, that sometimes the matrix H or $H^\top$ is used in the transformation, and sometimes H^{-1}.

These three examples of coordinate transformations may be written as follows:

$$\begin{aligned} \hat{q}^i &= (H^{-1})^i_j q^j && \text{(B.1)} \\ \hat{\lambda}_i &= H^j_i \lambda_j && \text{(B.2)} \\ \hat{P}^i_j &= (H^{-1})^i_k G^l_j P^k_l && \text{(B.3)} \end{aligned}$$

where we use the tensor summation convention that an index repeated in upper and lower positions in a product represents summation over the range of the index. Note that those indices that are written as superscripts transform according to H^{-1}, whereas those that are written as subscripts transform as H (or G). Note that there is no distinction in tensor notation between indices that are transformed by H, and those that are transformed by $H^\top$. In general, tensor indices will transform by either H or H^{-1}. Those indices that transform according to H are known as *covariant* indices and are written as subscripts (Hartley, 1997a). Those indices that transform according to H^{-1} are known as *contravariant* indices, and are written as superscripts (Hartley,

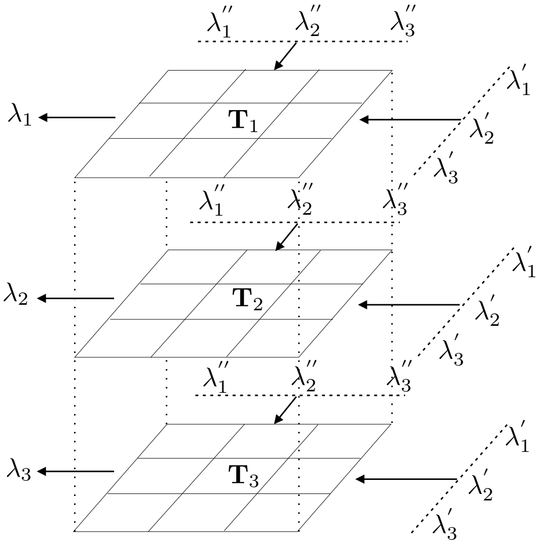


Figure B.1: 3D representation of the trifocal tensor. The picture represents $\lambda_i = \lambda_j^{'}\lambda_k^{''}\mathcal{T}_i^{jk}$, which is the contraction of the tensor with the lines $\boldsymbol{\lambda}'$ and $\boldsymbol{\lambda}''$ to produce a line $\boldsymbol{\lambda}$. In pseudo-matrix notation this can be written as $\lambda_i = \boldsymbol{\lambda}^{'\top} T_i \boldsymbol{\lambda}^{''\top}$, where $(T_i)_{jk} = \mathcal{T}_i^{jk}$ (Hartley and Zisserman, 2003).

1997a). The number of indices is the *valency* of the tensor. The sum over an index, e.g. $H_i^j\lambda_j$, is referred to as a *contraction*, in this case the tensor H_i^j is contracted with the line λ_j.

B.1 Tensorial notation and the trifocal tensor

The trifocal tensor $\mathcal{T}_i^{jk}$ has one covariant and two contravariant indices. For vectors and matrices, such as q^i, λ_i and P_j^i, it is possible to write the transformation rules using standard linear algebra notation, e.g. $\mathbf{q}^{'} = H\mathbf{q}$. However, for tensors with three or more indices, this cannot conveniently be done. A vector $\mathbf{q}$ may be thought of as a set of numbers arranged in a column or row, and a matrix H as a 2D array of numbers. Similarly, a tensor with three indices may be thought of as a 3D array of numbers. In particular the trifocal tensor is a $3 \times 3 \times 3$ cube of cells as illustrated in Figure B.1.

Appendix C

Multiple View Geometry

C.1 Three-View Geometry

The geometry of three perspective views may be acquired simultaneously as in a trinocular rig, or acquired sequentially, for example by a camera moving relative to the scene. Exactly as in the case of two-view geometry, we say that these two situations are geometrically equivalent and they will not be differentiated here.

A new multiple view object – the *trifocal tensor* – plays an analogous role in three views to that played by the fundamental matrix in two views (Hartley, 1997b). It encapsulates all the (projective) geometric relations between three views that are independent of scene structure. In the following we present its derivation as well as some of its main properties.

C.1.1 The Trifocal Tensor $\mathcal{T}$

The trifocal tensor may be approached in several different manners (Shashua and Werman, 1995; Hartley, 1997b; Hartley and Zisserman, 2003; Canterakis, 2000), but in this section we will present its derivation based on the description performed by Hartley (1997b), where the starting point is taken to be the incidence relationship of three corresponding lines. Finally, we will present the application of the trifocal tensor as tool for transferring points.

Essentially, the trifocal tensor is a triply indexed 3×3×3 array of values. Therefore, it is fully natural to treat it as a tensor (Viéville and Luong, 1993). In this section we will make use of tensorial notation, in particular the standard summation convention for repeated indices. Some basics on tensorial notation are presented in Appendix B.

Line Transfer

Consider the three cameras with image planes $\boldsymbol{\pi}^{'}$, $\boldsymbol{\pi}^{''}$ and $\boldsymbol{\pi}^{'''}$ in Figure C.1. Their corresponding projection matrices will be denoted by $\mathcal{M}^{'} = [I|0]$, $\mathcal{M}^{''} = [a_i^j]$ and $\mathcal{M}^{'''} = [b_i^j]$. Observe that we are picking the coordinate system attached to the camera $\boldsymbol{\pi}^{'}$ as the world reference frame. Let $\boldsymbol{\lambda}^{'}$, $\boldsymbol{\lambda}^{''}$ and $\boldsymbol{\lambda}^{'''}$ be the image lines of the scene line illustrated in Figure C.1. Their corresponding planes in space are given by $\boldsymbol{\varphi}^{'} = \mathcal{M}^{'\top}\boldsymbol{\lambda}^{'}$, $\boldsymbol{\varphi}^{''} = \mathcal{M}^{''\top}\boldsymbol{\lambda}^{''}$ and $\boldsymbol{\varphi}^{'''} = \mathcal{M}^{'''\top}\boldsymbol{\lambda}^{'''}$. Our goal here is to find a relationship between the coordinates of these three lines.

Given that the three image lines are derived from a single line in space, it follows that $\boldsymbol{\varphi}^{'}$, $\boldsymbol{\varphi}^{''}$ and $\boldsymbol{\varphi}^{'''}$ must meet at this line in space. This fact leads to a linear dependency between the coordinates of these three planes. In particular, there exist constants ρ_1 and ρ_2 such that we have

$$\boldsymbol{\varphi}^{'} = \rho_1\boldsymbol{\varphi}^{''} + \rho_2\boldsymbol{\varphi}^{'''}. \tag{C.1}$$

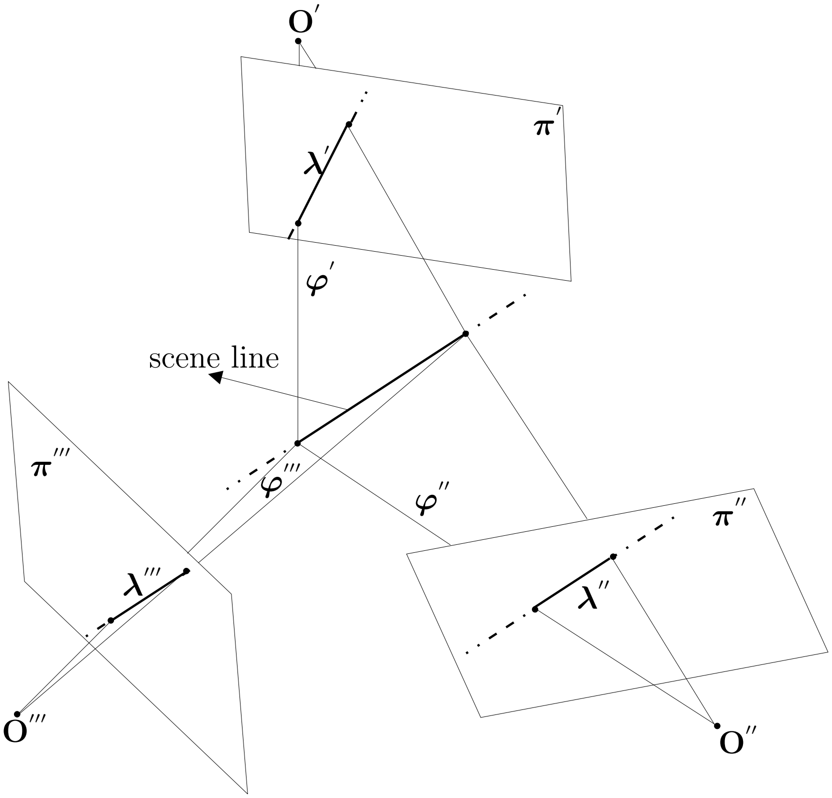


Figure C.1: Three images of a line define it as the intersection of three planes in the same pencil.

Writing Equation (C.1) in terms of the planes' coordinates we obtain

$$\lambda_i^{'} = \rho_1 \left(a_i^j \lambda_j^{''} \right) + \rho_2 \left(b_i^k \lambda_k^{'''} \right), \qquad (i = 1, ..., 3) \tag{C.2}$$

$$0 = \rho_1 \left(a_4^j \lambda_j^{''} \right) + \rho_2 \left(b_4^k \lambda_k^{'''} \right). \tag{C.3}$$

From Equation (C.3) we deduce that $\rho_1 \approx (b_4^k \lambda_k^{'''})$ and $\rho_2 \approx -(a_4^j \lambda_j^{''})$. The notation $\approx$ denotes equality up to an unknown scale factor. Thus, we may rewrite Equation (C.2) as

$$\begin{aligned} \lambda_i^{'} &\approx (b_4^k \lambda_k^{'''})(a_i^j \lambda_j^{''}) - (a_4^j \lambda_j^{''}) b_i^k \lambda_k^{'''}), && \text{(C.4)} \\ &= \lambda_j^{''} \lambda_k^{'''} (a_i^j b_4^k - a_4^j b_i^k). && \text{(C.5)} \end{aligned}$$

If we define a 3×3×3 tensor $\mathcal{T}_i^{jk}$ by the expression

$$\mathcal{T}_i^{jk} = a_i^j b_4^k - a_4^j b_i^k, \tag{C.6}$$

we obtain the following equation

$$\lambda_i^{'} \approx \lambda_j^{''} \lambda_k^{'''} \mathcal{T}_i^{jk}. \tag{C.7}$$

The tensor $\mathcal{T}_i^{jk}$ is the so-called *trifocal tensor* (Hartley and Zisserman, 2003) and may be thought of as a set of three 3×3 matrices of rank 2, which is evident from Equation (C.6), where it is expressed as the sum of two outer products. Given $\mathcal{T}_i^{jk}$ and the coordinates $\lambda_j^{''}$, $\lambda_k^{'''}$ of corresponding lines, Equation (C.7) may be used to compute the line in the other image. This process is called *line transfer.*

Importantly, if at least 13 line matches are known, it is possible to solve for the entries of the tensor $\mathcal{T}_i^{jk}$, since each line match provides two linear equations in the 27 unknown tensor entries. In particular, if the line $\boldsymbol{\lambda}^{'}$ is specified by two points on the line, then each such point $\mathbf{q}^{'} = (q^{'1}, q^{'2}, q^{'3})$ generates an equation such as

$$q^{'i} \lambda_j^{''} \lambda_k^{'''} \mathcal{T}_i^{jk} = 0. \tag{C.8}$$

Point Transfer

Suppose that a scene point $\mathbf{Q}$ is seen at positions $\mathbf{q}^{'}$, $\mathbf{q}^{''}$ and $\mathbf{q}^{'''}$ in three image planes $\boldsymbol{\pi}^{'}$, $\boldsymbol{\pi}^{''}$ and $\boldsymbol{\pi}^{'''}$, respectively, where $\mathbf{q}^{'}$ (and similarly $\mathbf{q}^{''}$ and $\mathbf{q}^{'''}$) are represented in homogeneous coordinates. Now, we wish to find a relationship between the coordinates of those three image points. It is important to say that at any point in the following derivation we may set the coordinates $q^{'3}$, $q^{''3}$ and $q^{'''3}$ to 1 to obtain equations relating to measured image coordinates.

Consider that the cameras' projection matrices are denoted by $\mathcal{M}^{'} = [I|0]$, $\mathcal{M}^{''} = [a_i^j]$ and $\mathcal{M}^{'''} = [b_i^j]$. Since $\mathbf{q}^{'} \approx [I|0]\mathbf{Q}$, we have

$$\mathbf{Q} = \begin{bmatrix} \mathbf{q}^{'} \\ w \end{bmatrix}, \tag{C.9}$$

for some w yet to be determined.

Therefore, projecting the scene point $\mathbf{Q}$ in the image plane $\boldsymbol{\pi}^{''}$ by the usual formula $q^{''i} \approx a_j^i Q^j$, we have

$$q^{''i} \approx a_k^i q^{'k} + a_4^i w. \tag{C.10}$$

As performed by Hartley (1997b), we may eliminate this scale factor to obtain equations

$$q^{''i}\left(a_k^j q^{'k} + a_4^j w\right) = q^{''j}\left(a_k^i q^{'k} + a_4^i w\right). \tag{C.11}$$

Each choice of the free indices i and j gives a separate equation. Of the three resulting equations, only two are independent. Thus, we obtain three separate estimates for w as follows

$$w = \frac{q'^{k}\left(q''^{i} a^{j}_{k} - q''^{j} a^{i}_{k}\right)}{q''^{j} a^{i}_{4} - q''^{i} a^{j}_{4}}. \tag{C.12}$$

Substituting w in Equation (C.9) by its value in Equation (C.12) we can write the scene point $\mathbf{Q}$ as

$$\mathbf{Q} = \left[\begin{array}{c} \mathbf{q}' \\ \dfrac{q'^{k}\left(q''^{i} a^{j}_{k} - q''^{j} a^{i}_{k}\right)}{q''^{j} a^{i}_{4} - q''^{i} a^{j}_{4}} \end{array}\right]. \tag{C.13}$$

Finally, projecting the scene point $\mathbf{Q}$ in the image plane $\boldsymbol{\pi}'''$ ($q'''^{l} \approx b^{l}_{k} Q^{k}$), we obtain

$$\begin{aligned} q'''^{l} &\approx b^{l}_{k} q'^{k}\left(q''^{j} a^{i}_{4} - q''^{i} a^{j}_{4}\right) + b^{l}_{4} q'^{k}\left(q''^{i} a^{j}_{k} - q''^{j} a^{i}_{k}\right) & \text{(C.14)} \\ &\approx q'^{k} q''^{i}\left(a^{j}_{k} b^{l}_{4} - a^{j}_{4} b^{l}_{k}\right) - q'^{k} q''^{j}\left(a^{i}_{k} b^{l}_{4} - a^{i}_{4} b^{l}_{k}\right). & \text{(C.15)} \end{aligned}$$

Observe that the tensor coefficients $\mathcal{T}^{jk}_{i}$ can be easily identified in Equation (C.15):

$$q'''^{l} \approx q'^{k}\left(q''^{i} \mathcal{T}^{jl}_{k} - q''^{j} \mathcal{T}^{il}_{k}\right). \tag{C.16}$$

As before we may eliminate the unknown scale factor to obtain the equations

$$q'^{k}\left(q''^{i} q'''^{l} \mathcal{T}^{jm}_{k} - q''^{j} q'''^{l} \mathcal{T}^{im}_{k} - q''^{i} q'''^{m} \mathcal{T}^{jl}_{k} + q''^{j} q'''^{m} \mathcal{T}^{il}_{k}\right) = 0^{ijlm}. \tag{C.17}$$

As mentioned by Hartley (1997b), these are the trilinearity relationships of Shashua and Werman (1995). The indices i, j, l and m are free variables, and there is one equation for each choice of indices with $i \neq j$ and $l \neq m$. We may assume that $i < j$ and $l < m$, since we have the same relation by interchanging i and j, or l and m. Therefore, from the Expression (C.17) we obtain nine possible equations. As only two of the three choices of pair (i,j) give independent equations, and the same is true for pairs (l,m), we get only four linearly independent equations.

One natural choice of the four independent equations from Expression (C.17) is obtained by setting $j = m = 3$, and letting i and l range freely, given the conditions stated before ($i \neq j$, $l \neq m$, $i < j$ and $l < m$). Thus, if we set $q^{'3}$, $q^{''3}$ and $q^{'''3}$ to 1, we obtain the relationship between image coordinates in Equation (C.18)

$$q^{'k}\left(q^{''i}q^{'''l}\mathcal{T}_k^{33} - q^{'''l}\mathcal{T}_k^{i3} - q^{''i}\mathcal{T}_k^{3l} + \mathcal{T}_k^{il}\right) = 0^{i3l3} \qquad (i, l = 1, 2). \qquad \text{(C.18)}$$

Equations (C.8) and (C.18) show the presence of the entries of the trifocal tensor $\mathcal{T}$ involved in the processes of transferring lines and points. In particular, each line correspondence provides two linear constraints on the entries $\mathcal{T}_i^{jk}$, whereas each point correspondence provides four linear equations. Since $\mathcal{T}$ has 27 entries, 26 equations are needed to solve for the $\mathcal{T}_i^{jk}$ up to scale, that is, provided that $2\ \#lines + 4\ \#points \geq 26$ (Hartley and Zisserman, 2003) we have enough matches to completely determine the entries of the trifocal tensor. The main methods for computing trifocal tensors (Hartley, 1997b; Torr and Zisserman, 1997; Faugeras and Papadopoulo, 1997) are based on that restriction and most of them are described in detail by Hartley and Zisserman (2003).

C.2 N-View Geometry

In the study of the geometry of multiple views, the fundamental matrix and the trifocal tensor have proven to be essential tools. In the case of four views, the quadrifocal tensor was proposed as a natural extension of those techniques (Heyden, 1995a; Hartley, 1998). Because of the added stability of a fourth view, use of the quadrifocal tensor should lead to greater accuracy than two and three-view techniques (Heyden, 1995a). However, the four-view tensor has not been given much attention in the literature (Hartley and Zisserman, 2003). One of the main reasons to its rare use is related to its over-parametrization, where 81 components of the tensor are used to describe a geometric configuration that depends only on 29 parameters (Hartley, 1998). This fact can lead to significant inaccuracies if additional constraints are not applied.

With regard to the general case of multiple images taken by N different cameras, several different approaches have been proposed in the literature (Luong and Vieville, 1994; Faugeras and Mourrain, 1995; Heyden and Åström, 1997; Heyden, 1998). In Heyden (1998), a common framework was proposed for the definition and operations on the different multiple view tensors. The author showed that there are essentially three different ways to encode an N-view geometry, namely by using bifocal tensors or fundamental matrices, trifocal tensors and quadrifocal tensors (combinations of these three tools are also possible).

According to his work, if only bifocal tensors are used for a sequence of temporally corresponding images, it is sufficient to use the tensors ${}_{i,i+1}\mathcal{F}$ and ${}_{i,i+2}\mathcal{F}$ for every triplet $({}_{i,i+1}\mathcal{F}, {}_{i,i+2}\mathcal{F}, {}_{i+1,i+2}\mathcal{F})$, where i denotes the i-th view and ${}_{i,i+1}\mathcal{F}$ denotes the fundamental matrix between views i and $i+1$. Using this representation and remembering that each bifocal tensor has 7 independent parameters (Hartley and Zisserman, 2003), we have $N-1+N-2=2N-3$ bifocal tensors and $N-2$ such triplets obeying three constraints each, giving in total $7(2N-3)-3(N-2)=11N-15$ parameters describing the N-view geometry, i.e. the minimal number (Heyden, 1998).

On the other hand, if only trifocal tensors are used it is sufficient to use the tensors ${}^{i+1,i+2}_{i}\mathcal{T}$. Using this representation we have $N-2$ trifocal tensors with 18 independent parameters each and for every consecutive pair $({}^{i+1,i+2}_{i}\mathcal{T}, {}^{i+2,i+3}_{i+1}\mathcal{T})$ of trifocal tensors we get 7 constraints from the compatibility of ${}_{i+1,i+2}\mathcal{F}$, that can be calculated from both. Since there are $N-3$ such constraints, we get $18(N-2)-7(N-3) = 11N-15$ parameters describing the N-view geometry, i.e. the minimal number (Heyden, 1998).

Finally, when only quadrifocal tensors (represented by letter $\mathcal{Q}$) are used it is sufficient to use the tensors ${}^{i,i+1,i+2,i+3}\mathcal{Q}$. Now, by using this representation we have $N-3$ quadrifocal tensors with 29 independent parameters each and for every consecutive pair $({}^{i,i+1,i+2,i+3}\mathcal{Q}, {}^{i+1,i+2,i+3,i+4}\mathcal{Q})$ of quadrifocal tensors we get 18 constraints from the compatibility of ${}^{i+2,i+3}_{i+1}\mathcal{T}$, that can be calculated from both. Since there are $N-4$ such constraints, we get $29(N-3)-18(N-4) = 11N-15$ parameters describing the N-view geometry, i.e the minimal number (Heyden, 1998).

An enumeration of the complete set of multilinear relations, formulae for the multiview tensors, and the analysis of the number of independent equations derived from point correspondences can be found in Hartley and Zisserman (2003) and Heyden (1998).