Method 1: Geometric Error Optimization

- we need to encode the constraints \( \hat{y}_i F \hat{x}_i = 0 \), rank \( F = 2 \)
- idea: reconstruct 3D point via equivalent projection matrices and use reprojection error
- equivalent projection matrices are see [H&Z, Sec. 9.5] for complete characterization

\[
P_1 = \begin{bmatrix} I & 0 \end{bmatrix}, \quad P_2 = \begin{bmatrix} [e_2]_x F + e_2 e_1^\top e_2 \end{bmatrix}
\]  

\( H3; 2pt: \) Assuming \( e_1, e_2 \) are epipoles of \( F \), verify that \( F \) is a fundamental matrix of \( P_1, P_2 \).

Hint: \( A \) is skew symmetric iff \( x^\top A x = 0 \) for all vectors \( x \).

1. compute \( F^{(0)} \) by the 7-point algorithm \( \rightarrow 83 \); construct camera \( P_2^{(0)} \) from \( F^{(0)} \) using (17)

2. triangulate 3D points \( \hat{X}_i^{(0)} \) from matches \( (x_i, y_i) \) for all \( i = 1, \ldots, k \) \( \rightarrow 88 \)

3. starting from \( P_2^{(0)}, \hat{X}_i^{(0)} \) minimize the reprojection error (15)

\[
(\hat{X}^*, P_2^*) = \arg \min_{P_2, \hat{X}} \sum_{i=1}^{k} e_i^2(\hat{Z}_i | \hat{Z}_i(\hat{X}_i, P_2))
\]

where

\[
\hat{Z}_i = (\hat{x}_i, \hat{y}_i) \quad \text{(Cartesian)}, \quad \hat{x}_i \simeq P_1 \hat{X}_i, \quad \hat{y}_i \simeq P_2 \hat{X}_i \quad \text{(homogeneous)}
\]

Non-linear, non-convex problem

4. compute \( F \) from \( P_1, P_2^* \)

- \( 3k + 12 \) parameters to be found: latent: \( \hat{X}_i \), for all \( i \) (correspondences!), non-latent: \( P_2 \)
- minimal representation: \( 3k + 7 \) parameters, \( P_2 = P_2(F') \) \( \rightarrow 143 \)
- there are pitfalls; this is essentially bundle adjustment; we will return to this later \( \rightarrow 134 \)
Given $P_1, P_2$ and a correspondence $x \leftrightarrow y$, look for 3D point $X$ projecting to $x$ and $y$. → 88

Idea:

1. if not given, compute $F$ from $P_1, P_2$, e.g. $F = (Q_1Q_2^{-1})^\top [q_1 - (Q_1Q_2^{-1})q_2]$

2. correct measurement by the linear estimate of the correction vector → 99

\[
\begin{bmatrix}
\hat{u}^1 \\
\hat{v}^1 \\
\hat{u}^2 \\
\hat{v}^2
\end{bmatrix} \approx \begin{bmatrix}
u^1 \\
v^1 \\
u^2 \\
v^2
\end{bmatrix} - \frac{\varepsilon}{\|J\|^2} J^\top = \begin{bmatrix}
u^1 \\
v^1 \\
u^2 \\
v^2
\end{bmatrix} - \frac{y^\top Fx}{\|SFx\|^2 + \|SF^\top y\|^2}\begin{bmatrix}(F_1)^\top y \\
(F_2)^\top y \\
(F_1)^\top x \\
(F_2)^\top x
\end{bmatrix}
\]

3. use the SVD triangulation algorithm with numerical conditioning → 89; iteration possible

Ex (cont’d from → 92):

- $S = \text{distf} (\gamma, \gamma, \delta)$
- $X_T$ – noiseless ground truth position
- $X_s$ – Sampson-corrected algebraic error minimizer
- $X_a$ – algebraic error minimizer
- $m$ – measurement ($m_T$ with noise in $v^2$)
Goal: Given a set $X = \{(x_i, y_i)\}_{i=1}^k$ of $k \gg 7$ inlier correspondences, compute a statistically efficient estimate for fundamental matrix $F$.

What we have so far

- 7-point algorithm for $F$ (5-point algorithm for $E$) $\rightarrow 83$
- definition of Sampson error per correspondence $e_i(F \mid x_i, y_i) \rightarrow 103$

What we need

- an optimization algorithm for $F^* = \arg \min_F \sum_{i=1}^k e_i^2(F \mid X)$
- the 7-point estimate is a good starting point $F_0$
Levenberg-Marquardt (LM) Iterative Estimation in a Nutshell

Consider error function $e_i(\theta) = f(x_i, y_i, \theta) \in \mathbb{R}^m$, with $x_i, y_i$ given, $\theta \in \mathbb{R}^q$ unknown

Our goal: $\theta^* = \arg \min_{\theta} \sum_{i=1}^k \|e_i(\theta)\|^2$

Idea 1 (Gauss-Newton approximation): proceed iteratively for $s = 0, 1, 2, \ldots$

$$\theta^{s+1} := \theta^s + d_s,$$

where

$$d_s = \arg \min_d \sum_{i=1}^k \|e_i(\theta^s + d)\|^2$$

$e_i(\theta^s + d) \approx e_i(\theta^s) + L_i d,$

$$(L_i)_{jl} = \frac{\partial (e_i(\theta))}{\partial (\theta)_l}, \quad L_i \in \mathbb{R}^{m,q}$$

typically a long matrix

Then the solution to Problem (19) is a set of normal eqs

$$- \sum_{i=1}^k L_i^\top e_i(\theta^s) = \left( \sum_{i=1}^k L_i^\top L_i \right) d_s,$$

$d_s = -L \backslash e$

$d_s$ can be solved for by Gaussian elimination using Choleski decomposition of $L$

$L$ symmetric $\Rightarrow$ use Choleski, almost $2 \times$ faster than Gauss-Seidel, see bundle adjustment

such updates do not lead to stable convergence $\rightarrow$ ideas of Levenberg and Marquardt
Idea 2 (Levenberg): replace $\sum_i L_i^T L_i$ with $\sum_i L_i^T L_i + \lambda I$ for some damping factor $\lambda \geq 0$

Idea 3 (Marquardt): replace $\lambda I$ with $\lambda \sum_i \text{diag}(L_i^T L_i)$ to adapt to local curvature:

$$-\sum_{i=1}^{k} L_i^T e_i(\theta^s) = \left( \sum_{i=1}^{k} (L_i^T L_i + \lambda \text{diag}(L_i^T L_i)) \right) d_s$$

Idea 4 (Marquardt): adaptive $\lambda$

1. choose $\lambda \approx 10^{-3}$ and compute $d_s$
2. if $\sum_i \| e_i(\theta^s + d_s) \|^2 < \sum_i \| e_i(\theta^s) \|^2$ then accept $d_s$ and set $\lambda := \lambda/10$, $s := s + 1$
3. otherwise set $\lambda := 10\lambda$ and recompute $d_s$

- sometimes different constants are needed for the 10 and $10^{-3}$
- note that $L_i \in \mathbb{R}^{m,q}$ (long matrix) but each contribution $L_i^T L_i$ is a square singular $q \times q$ matrix (always singular for $k < q$)
- error can be made robust to outliers, see the trick $\rightarrow 111$
- we have approximated the least squares Hessian by ignoring second derivatives of the error function (Gauss-Newton approximation)  
  See [Triggs et al. 1999, Sec. 4.3]
- $\lambda$ helps avoid the consequences of gauge freedom $\rightarrow 139$
- modern variants of LM are Trust Region methods
**LM with Sampson Error for Fundamental Matrix Estimation**

**Sampson** (derived by linearization over point coordinates $u^1, v^1, u^2, v^2$)

$$e_i(F) = \frac{\varepsilon_i}{\|J_i\|} = \frac{y_i^\top F x_i}{\sqrt{\|S F x_i\|^2 + \|S F^\top y_i\|^2}}$$

where $S = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

**LM** (by linearization over parameters $F$)

$$L_i = \frac{\partial e_i(F)}{\partial F} = \cdots = \frac{1}{2\|J_i\|} \left[ \left( y_i - \frac{2e_i}{\|J_i\|} S F x_i \right) x_i^\top + y_i \left( x_i - \frac{2e_i}{\|J_i\|} S F^\top y_i \right) \right]^\top$$ (21)

- $L_i$ in (21) is a $3 \times 3$ matrix, must be reshaped to dimension-9 vector $\text{vec}(L_i)$ to be used in LM
- $x_i$ and $y_i$ in Sampson error are normalized to unit homogeneous coordinate (21) relies on this
- reinforce rank $F = 2$ after each LM update to stay in the fundamental matrix manifold and $\|F\| = 1$ to avoid gauge freedom by SVD →109
- LM linearization could be done by numerical differentiation (small dimension)
Local Optimization for Fundamental Matrix Estimation

Given a set $X = \{(x_i, y_i)\}_{i=1}^k$ of $k \gg 7$ inlier correspondences, compute a statistically efficient estimate for fundamental matrix $F$.

Summary so far

1. Find the conditioned (→91) 7-point $F_0$ (→83) from a suitable 7-tuple
2. Improve the $F_0^*$ using the LM optimization (→106–107) and the Sampson error (→108) on all inliers, reinforce rank-2, unit-norm $F_k^*$ after each LM iteration using SVD

We are not yet done

- if there are no wrong correspondences (mismatches, outliers), this gives a local optimum given the 7-point initial estimate
- the algorithm breaks under contamination of (inlier) correspondences by outliers
- the full problem involves finding the inliers!
- in addition, we need a mechanism for jumping out of local minima (and exploring the space of all fundamental matrices)
The Full Problem of Matching and Fundamental Matrix Estimation

Problem: Given image point sets $X = \{x_i\}_{i=1}^m$ and $Y = \{y_j\}_{j=1}^n$ and their descriptors $D$, find the most probable

1. inliers $S_X \subseteq X$, $S_Y \subseteq Y$
2. one-to-one perfect matching $M : S_X \rightarrow S_Y$
3. fundamental matrix $F$ such that $\text{rank } F = 2$
4. such that for each $x_i \in S_X$ and $y_j = M(x_i)$ it is probable that
   a) the image descriptor $D(x_i)$ is similar to $D(y_j)$, and
   b) the total geometric error $E = \sum_{i,j} e_{ij}^2(F)$ is small
5. inlier-outlier and outlier-outlier matches are improbable

\[
(M^*, F^*) = \arg \max_{M, F} p(E, D, F | M) P(M) \tag{22}
\]

- probabilistic model: an efficient language for problem formulation
- the (22) is a Bayesian probabilistic model
- binary matching table $M_{ij} \in \{0, 1\}$ of fixed size $m \times n$
  - each row/column contains at most one unity
  - zero rows/columns correspond to unmatched point $x_i/y_j$
- it also unifies 4.a and 4.b
- there is a constant number of random variables!
Deriving A Robust Matching Model by Approximate Marginalization

For algorithmic efficiency, instead of $(M^*, F^*) = \arg \max_{M,F} p(E, D, F | M) P(M)$ solve

$$F^* = \arg \max_F p(E, D, F)$$

by marginalization of $p(E, D, F | M) P(M)$ over $M$

this changes the problem!

ignoring that $M$ are 1:1 matchings and assuming correspondence-wise independence:

$$p(E, D, F | M)P(M) = \prod_{i=1}^{m} \prod_{j=1}^{n} p_e(e_{ij}, d_{ij}, F | m_{ij})P(m_{ij})$$

- $e_{ij}$ represents geometric error for match $x_i \leftrightarrow y_i$: $e_{ij}(x_i, y_i, F)$
- $d_{ij}$ represents descriptor similarity for match $x_i \leftrightarrow y_i$: $d_{ij} = \|d(x_i) - d(y_j)\|$ 

Marginalization:

$$p(E, D, F) \approx \sum_{m_{11} \in \{0,1\}} \sum_{m_{12}} \cdots \sum_{m_{mn}} p(E, D, F | M)P(M) =$$

$$= \sum_{m_{11}} \cdots \sum_{m_{mn}} \prod_{i=1}^{m} \prod_{j=1}^{n} p_e(e_{ij}, d_{ij}, F | m_{ij})P(m_{ij}) = \otimes 1 =$$

$$= \prod_{i=1}^{m} \prod_{j=1}^{n} \sum_{m_{ij} \in \{0,1\}} p_e(e_{ij}, d_{ij}, F | m_{ij})P(m_{ij})$$

we will continue with this term
Robust Matching Model (cont’d)

\[
\sum_{m_{ij} \in \{0,1\}} p_e(e_{ij}, d_{ij}, F | m_{ij}) P(m_{ij}) =
\]

\[
= \underbrace{p_e(e_{ij}, d_{ij}, F | m_{ij} = 1)}_{p_1(e_{ij}, d_{ij}, F)} P(m_{ij} = 1) + \underbrace{p_e(e_{ij}, d_{ij}, F | m_{ij} = 0)}_{p_0(e_{ij}, d_{ij}, F)} P(m_{ij} = 0) =
\]

\[
= (1 - P_0) p_1(e_{ij}, d_{ij}, F) + P_0 p_0(e_{ij}, d_{ij}, F)
\]  
(24)

- the \( p_0(e_{ij}, d_{ij}, F) \) is a penalty for ‘missing a correspondence’ but it should be a p.d.f. (cannot be a constant)  
  (→113 for a simplification)

choose \( P_0 \to 1, \ p_0(\cdot) \to 0 \) so that \( \frac{P_0}{1 - P_0} p_0(\cdot) \approx \text{const} \)

- the \( p_1(e_{ij}, d_{ij}, F) \) is typically an easy-to-design term: assuming independence of geometric error and descriptor similarity:

\[
p_1(e_{ij}, d_{ij}, F) = p_1(e_{ij} | F) p_F(F) p_1(d_{ij})
\]

- we choose, eg.

\[
p_1(e_{ij} | F) = \frac{1}{Te(\sigma_1)} e^{-\frac{e_{ij}^2(F)}{2\sigma_1^2}}, \quad p_1(d_{ij}) = \frac{1}{T_d(\sigma_d, \dim d)} e^{-\frac{\|d(x_i) - d(y_j)\|^2}{2\sigma_d^2}}
\]  
(25)

- \( F \) is a random variable and \( \sigma_1, \sigma_d, P_0 \) are parameters
- the form of \( T(\sigma_1) \) depends on error definition, it may depend on \( x_i, y_j \) but not on \( F \)
- we will continue with the result from (24)
Simplified Robust Energy (Error) Function

• assuming the choice of $p_1$ as in (25), we are simplifying

$$p(E, D, F) = p(E, D | F) p_F(F) = p_F(F) \prod_{i=1}^m \prod_{j=1}^n \left[(1 - P_0) p_1(e_{ij}, d_{ij} | F) + P_0 p_0(e_{ij}, d_{ij} | F)\right]$$

• we choose $\sigma_0 \gg \sigma_1$ and omit $d_{ij}$ for simplicity; then the square-bracket term is

$$\frac{1 - P_0}{T_e(\sigma_1)} e^{-\frac{e_{ij}^2(F)}{2\sigma_1^2}} + \frac{P_0}{T_e(\sigma_0)} e^{-\frac{e_{ij}^2(F)}{2\sigma_0^2}}$$

• we define the ‘potential function’ as: $V(x) = -\log p(x)$, then

$$V(E, D | F) = \sum_{i=1}^m \sum_{j=1}^n \left[-\log \frac{1 - P_0}{T_e(\sigma_1)} - \log \left(e^{-\frac{e_{ij}^2(F)}{2\sigma_1^2}} + \frac{P_0}{1 - P_0} \frac{T_e(\sigma_1)}{T_e(\sigma_0)} e^{-\frac{e_{ij}^2(F)}{2\sigma_0^2}}\right) \Delta \approx \text{const}\right] =$$

$$= mn \Delta + \sum_{i=1}^m \sum_{j=1}^n -\log \left(e^{-\frac{e_{ij}^2(F)}{2\sigma_1^2}} + t\right) \hat{V}(e_{ij})$$

• note we are summing over all $mn$ matches ($m$, $n$ are constant)
The Action of the Robust Matching Model on Data

Example for $\hat{V}(e)$ from (26):

- red – the usual (non-robust) error
- blue – the rejected correspondence penalty $t$
- green – ‘robust energy’ (26)

- if the error of a correspondence exceeds a limit, it is ignored
- then $\hat{V}(e) = \text{const}$ and we essentially count outliers in (26)
- $t$ controls the ‘turn-off’ point
- the inlier/outlier threshold is $e_T$ – the error for which $(1 - P_0) p_1(e_T) = P_0 p_0(e_T)$: note that $t \approx 0$

$$e_T = \sigma_1 \sqrt{- \log t^2}$$ (27)

The full optimization problem (23) uses (26):

$$F^* = \arg \max_F \left\{ \frac{p(E, D \mid F) \cdot p(F)}{p(E, D)} \right\} \approx \arg \min_F \left[ V(F) + \sum_{i=1}^{m} \sum_{j=1}^{n} \log \left( e - \frac{e_{ij}^2(F)}{2\sigma_1^2} + t \right) \right]$$

- typically we take $V(F) = -\log p(F) = 0$ unless we need to stabilize a computation, e.g. when video camera moves smoothly (on a high-mass vehicle) and we have a prediction for $F$
- evidence is not needed unless we want to compare different models (eg. homography vs. epipolar geometry)
Thank You
$C_1$

$e_1$

$m_a$,

$m_s$,

$m_T = m$