Note that S_{i,j_i} may be undefined, if the match was the result of a merge between several multi-frame matches. We consider it as a "don't care" set.

Define a relation $Q_{\overline{M}}$ over the multi-frame matches as follows:

For every pair of multi-frame matches $\overline{m}_1$ and $\overline{m}_2$; $Q_{\overline{M}}(\overline{m}_1, \overline{m}_2)$ is true *iff*

- $\overline{m}_1$ and $\overline{m}_2$ have at least two common frames, that is $\max(f_1, f_2) < \min(k_1, k_2)$, and
- Both multi-frame matches have the *same* equivalence sets $P_{i,j}$ for the common frames.

Note that $Q_{\overline{M}}$ is not an equivalence relation, because transitivity is not preserved.

Our partition of the multi-frame matches is based on ensuring that every multi-frame match $\overline{m}_1$ in a set will have at least one other multi-frame match $\overline{m}_2$ in this set such that $Q_{\overline{M}}(\overline{m}_1, \overline{m}_2)$ is true, and that no contradictions due to the non transitive nature of $Q_{\overline{M}}$ occur. Note that this partition may not be unique, as a different order of matches may yield somewhat different sets.

Our algorithm to partition the multi-frame matches is described in figure 5.3.1

Let $\overline{M}$ be the set of multi-frame matches.

An example for the initial partitioning is shown in figure 5.5.

The resulting partition of the multi-frame sets is a collection of sets of multi-frame matches with non-contradicting pairwise equivalence sets. Unfortunately, the two problems mentioned in the previous sections still exist, and in the next section we show our solution to them.

5.3.2 Splitting multi-frame matches sets

In this subsection we describe our method for detecting sets with several objects and how to separate them.

The problem of several different objects (or portions of objects) joined into one set is more difficult to resolve. We base our solution on the fact that if two objects were (partially) joined together, thus have similar 2-D motion and close proximity in some part of the sequence, but have different 3-D motions, then there have to be another part of the sequence where they cannot be joined, unless their 3-D motion

procedure $Q_{\overline{M}}$
While $\overline{M} \neq \emptyset$ **do**
    Select a multi-frame match $\overline{m}$ and assign it to a new multi-frame set $\overline{S}$.
    Remove $\overline{m}$ from $\overline{M}$.
    Repeat
        For every $\overline{m} \in \overline{M}$ **do**
            If there exists $\overline{m_1} \in \overline{S}$ such that $Q_{\overline{M}}(\overline{m}, \overline{m_1})$ is true and
              For every $\overline{m_2} \in S$ either
                $Q_{\overline{M}}(\overline{m}, \overline{m_2})$ is true or
                $\overline{m}$ and $\overline{m_2}$ share no common frames;
            then add $\overline{m}$ to $\overline{S}$ and remove it from $\overline{M}$.
        od
    until no new matches are added to $\overline{S}$
od

Figure 5.4: Multi-frame segmentation algorithm

differs so little as to render them indistinguishable. Therefore, these erroneous multi-frame sets would typically have many short multi-frame matches for the first or last frames, a few long or short multi-frame matches for the middle frames, but fairly small connectivity between the short multi-frame matches of the first frames and those of the last frames. (Note that "short" and "long" in this context refer to the number of frames participating in the match.h) Therefore separating the set into several subsets based on begin and end frame of each match, has a high likelihood of having most of the matches of one object in one pool and most of the matches of the other object in another pool.

Figure 5.6 illustrates the problem. Multi-frame section matches for the rectangle are divided between two segmented regions, because short multi-frame matches, representing motion through only 3 or 4 frames, were joined with multi-frame matches of the ellipse.

An example for this problem occurs in the sequence of the advancing car. In this sequence the observer is moving towards the car, and therefore the background is also deemed "moving." In the first several frames, the motion of the car is practically indistinguishable from that of the background. The algorithm therefore