SLAC-PUB-6349
September 1993
(T/E)

A Unified Treatment of Track Reconstruction and Particle Identification *

Richard Blankenbecler
Stanford Linear Accelerator Center
Stanford University, Stanford, California 94309

In this note, the deformable template or elastic arm approach to track reconstruction described previously will be extended to include particle identification. The discussion will first develop the mathematical and algorithmic structure of the pattern recognition problem using the methodology and language of statistical mechanics and then sketch its implementation in an object oriented programming language . This unified treatment will allow the user the freedom to organize the analysis in a variety of ways, with intermediate results available and documented.

I. INTRODUCTION

In this paper an extension of the elastic arms track reconstruction procedure given earlier [1] will be described that allows the algorithm to be applied to non-geometric quantities, in this case, to particle-type identification. This work was based on the work of Ohlsson, Peterson, and Yuille [2,3]. Another basic reference is the paper of Gyulassy and Harlander, [4] A more extensive list of relevant work can be found in these references.

While the present formalism is clearly applicable to general geometries, for reasons of simplicity and clarity, a small solid angle detector with NO magnetic field will serve as the "model" detector as in Ref. [1]. Certain correlations of measurements and fit parameters are also ignored. These assumptions simply serve to ease the burden of description.

We shall start by discussing track reconstruction in an ideal detector, add particle identification and noise to the system. The concept of "track" as used in ref [1] will be extended, subclassed in the parlance of object oriented programming (OOP), to include particle identification (ID) data. Among this additional information will be an array of probabilities that this track is of a given particle type or given particle type sequence, to be defined later. The meaningfulness of these probabilities depends upon the additional measurement information used in the fitting. The problem of ambiguities, or mirror charges, will be mentioned but the reader is referred to our treatment in ref [1] for a more complete description.

II. DETECTOR HAMILTONIAN

The deformable template or elastic arm approach to this problem starts by defining a cost function that is minimized when the assumed parametrized tracks fit the given hits. The same approach will be used to treat the problem of particle identification. To this end, and following the spirit of earlier references in which a statistical mechanical language was adopted, we first introduce a Hamiltonian function H for a *perfect* detector. This name change is to eliminate possible confusion between the theoretical simulation Hamiltonian or "cost" function and a true energy with all the accompanying connotations. The Hamiltonian is linear in the energy, or cost, of each track t ,

$$H_{\text{track}\leftarrow\text{hit}} = \sum_h \sum_t A_{th} M_{th} \ , \tag{1}$$

where M_{th} is a suitably chosen geometric measure of the "miss" such as the distance of closest approach divided by the hit resolution (or the square of this ratio). A_{th} is an assignment, or binary decision unit, such that $A_{th} = 1$ if hit h is associated with track t and is zero otherwise. The A's will be generalized and allowed to become fractional in the next section. Recall that each M_{th} depends upon the parameters of the track. The minimization of the Hamiltonian will eventually determine the track parameters.

*Work supported by Department of Energy contract DE–AC03–76SF00515.

The inclusion of noise is straightforward. The Hamiltonian is modified so that there is a penalty if a hit is *not* assigned to any of the tracks. One now writes

$$H_{\rm track\leftarrow hit} \;\; = \sum_h \left\{ \sum_t A_{th} M_{th} \;\; + \;\; \nu \left(1 - \sum_t A_{th} \right)^2 \right\} \;\; . \tag{2}$$

Thus the cost of classifying a hit as noise is ν.

Particle Identification: This problem consists of properly assigning a particular track to a definite particle type (or at least give probabilities of such assignments) based upon available discrimination data. The Hamiltonian is extended to include terms that not only assign hits to tracks (and thus depend upon the track parameters) but also to include terms that assign tracks to particle types. The new miss distance is not a geometric quantity, but a suitable miss in "discriminate space." The total Hamiltonian is then written as

$$H_{\rm total} \;\; = H_{\rm track\leftarrow hit} \;\; + \;\; H_{\rm pid\leftarrow track} \;\; , \tag{3}$$

where the second term is

$$H_{\rm pid\leftarrow track} \;\; = \sum_t \left\{ \sum_p A_{pt} M_{pt} \;\; + \;\; \rho \left(1 - \sum_p A_{pt} \right)^2 \right\} \;\; . \tag{4}$$

The assignment variable A_{pt} assigns the track t to the particle type p. The parameter ρ is the cost incurred when a track *cannot* be assigned to any listed particle type. What is the miss distance M_{pt} precisely? It is a function of all the additional experimental information that distinguishes between particle types. For example, it depends upon the differential energy loss, pulse heights along the track, Cherenkov measurements, penetration, etc., The detail structure of M_{pt} is, of course, detector dependent. It may be written in the general form

$$M_{pt} \;\; = \;\; M_{pt}\,[\mathrm{dEdx}] \;\; + \;\; M_{pt}\,[\mathrm{pulseheight}] + \;\; ... \tag{5}$$

The miss functions are functions of the difference between the measured and the expected values of a discriminate for a particle of type p divided by the resolution of the measurement. It is then natural to write for each discriminate d,

$$M_{pt}\,[d] \;\; = \;\; w_d \, * \sum_m \left(d(\mathrm{measured}) \;\; - \;\; d(\mathrm{expected\,for\,p}) \;\right)^2 / (\mathrm{res})^2 \;\; , \tag{6}$$

where w_d is the relative weight of this discriminate is the sum in Eq. 5 and the sum is over measurements along the track. For example, for differential energy loss measurements, $d \;=\; \frac{dE}{dx}$, Similarly, one may also choose the pulse height from hit h as a discriminate and restrict the sum to include only those hits *owned* by track t.

III. SIMULATED ANNEALING

Our mathematical problem is to find the global minimum of the Hamiltonian assuming that the number of tracks, etc., and estimates for their parameters are known. An efficient method for treating this problem is simulated annealing; the first step is to introduce a Boltzmann distribution for the assignment variables and the track parameters

$$P[\,\mathbf{A_H}\,,\,\mathbf{A_T}\,;\,\mathbf{T}\,] = \frac{1}{Z}\; e^{-\beta H[\,\mathbf{A_H}\,,\,\mathbf{A_T}\,;\,\mathbf{T}\,]} \;\; , \tag{7}$$

where $\mathbf{A_H}$ $(= \{A_{th}\})$ is the set of assignment variables assigning hits to tracks, $\mathbf{A_T}$ $(= \{A_{pt}\})$ assigns tracks to particle type, and $\mathbf{T}$ $(= \{T_t\})$ is the collection of parameters of all the tracks, where T_t is the set of parameters belonging to the track t. β is the inverse temperature which is introduced to pace the approach to the final fit. Finally, the partition function is normalized by summing over all allowed values of $\mathbf{A_H}$, $\mathbf{A_T}$ and $\mathbf{T}$

Now let us compute the *marginal* probability that describes the distribution of track parameters for a *uniform* distribution of assignments $\mathbf{A_H}$ and $\mathbf{A_T}$:

$$P[\,\mathbf{T}\,] = \sum_{A_{th}} \sum_{A_{pt}} P[A_{th}, A_{pt}; \mathbf{T}\,] \;\; \equiv \;\; \frac{1}{Z}\; e^{-\beta H_{\rm eff}\;[\,\mathbf{T}\,]} \;\; , \tag{8}$$

where an *effective* Hamiltonian has been defined that will prove convenient. If the distribution of assignments is not uniform, these prior probabilities enter as weights of the various terms in equation 8. It can be shown that for ideal data, the final fit parameters do not depend upon these weights.

The evaluation of the sums over the allowed values of $\mathbf{A_H}$ and $\mathbf{A_T}$ is straightforward. Assuming that the discriminate functions do *not* depend upon the $\mathbf{A_H}$, the marginal probability is then

$$P[\,\mathbf{T}\,] = \frac{1}{Z}\prod_h D_{tr}\,(h)\prod_t D_{id}\,(t)\ , \tag{9}$$

$$\text{where} \qquad D_{tr}\,(h) = \left\{ e^{-\beta\nu} + \sum_t e^{-\beta M_{th}} \right\} \tag{10}$$

$$\text{and} \qquad D_{id}\,(t)\ = \left\{ e^{-\beta\rho} + \sum_p e^{-\beta M_{pt}} \right\}\ . \tag{11}$$

The effective Hamiltonian is then

$$H_{\text{eff}}\ [\,\mathbf{T}\,] = -\frac{1}{\beta}\left\{ \sum_h \log D_{tr}\,(h)\ +\ \sum_t \log D_{id}\,(t) \right\} \tag{12}$$

$$= \sum_h H_{\text{eff}}\ [h,\,\mathbf{T}\,]\ +\ \sum_t H_{\text{eff}}\ [t,\,\mathbf{T}\,]\ . \tag{13}$$

Finally, note that each track t may have several parameters.

Minimization: We are looking for the most probable configurations, i.e. the number of tracks, their parameter values and the values of the assignment variables that minimize the cost in the limit of low "temperatures." In this limit of large β, the wrong assignment configurations, i.e. those with a finite miss distance M_{th}, are exponentially suppressed in the marginal probability. Hence in order to find the best fit to the hit data, we choose to minimize the effective cost function $H_{\text{eff}}\ [\,\mathbf{T}\,]$.

To avoid being trapped in a local minima, the solutions for the track parameters are computed and followed for a range of increasing values of β. Using the gradient descent method, at each stage in this iteration the parameters of each track t are changed by

$$\delta\,\overrightarrow{T}_t = -\eta\ \overrightarrow{\bigtriangledown}_{T_t} H_{\text{eff}}\ [\,\mathbf{T}\,] \tag{14}$$

$$= -\eta\ \sum_h\ \overrightarrow{\bigtriangledown}_{T_t} H_{\text{eff}}\ [h,\,\mathbf{T}\,] - \eta\ \overrightarrow{\bigtriangledown}_{T_t} H_{\text{eff}}\ [t,\,\mathbf{T}\,]\ . \tag{15}$$

The scalar parameter η is used to control the rate of approach to the minimum. The two terms in can have different η values. If T is a mixture of linear, angular, etc., variables, a tensor η is required as was discussed in ref [3].

Interpretation: An explicit enumeration of the derivatives in leads to a simple physical interpretation of this formulation. Introducing the effective assignment probabilities as in [1] leads to a simplification of the formulas. Note that these quantities have all the requisite properties of probabilities; whether or not they are actual physical probabilities depends upon the choices made for the miss functions, etc.

One introduces the "thermalized" assignment probability of hit h to track t as

$$< A_{th} > =\ \frac{e^{-\beta M_{th}}}{D_{tr}\,(h)}\ , \tag{16}$$

and the probability of assigning hit h to noise as

$$< \text{noise}(h) > =\ \frac{e^{-\beta\nu}}{D_{tr}\,(h)}\ , \tag{17}$$

with $\sum_t\ < A_{th} >\ +\ < \text{noise}(h) > =\ 1$. This simply states that a hit must be assigned either to one of the tracks or to noise. In addition, the probability that track t is of particle type p is defined as

$$< A_{pt} > =\ \frac{e^{-\beta M_{pt}}}{D_{id}\,(t)}\ , \tag{18}$$

while the probability ufo(t) that track t cannot be assigned to any particle type is

$$< \text{ufo}(t) > = \frac{e^{-\beta\rho}}{D_{id}\,(t)} \ , \tag{19}$$

with $\sum_p \ < A_{pt} > \ + \ < \text{ufo}(t) > = \ 1$. This simply states that a track must be assigned either to a listed particle type or declared unknown.

The gradient descent equations can now be written as

$$\delta \overrightarrow{T}_t \ = \ -\eta \ \sum_h < A_{th} > \ \overrightarrow{\bigtriangledown}_{T_t} M_{th} \ \ - \eta \ \sum_p < A_{pt} > \ \overrightarrow{\bigtriangledown}_{T_t} M_{pt} \ . \tag{20}$$

Thus it is seen that hits with large values of the assignment probabilities $< A_{th} >$ and particle types with large values of $< A_{pt} >$ dominate the determination of the parameters of track t .

Ambiguities: As discussed in ref [1], when ambiguities are present, one introduces the quantities

$$< a^+_{th} \ > = \ \frac{e^{-\beta M^+_{th}}}{e^{-\beta M^+_{th}} + e^{-\beta M^-_{th}}} \quad \text{and} \quad < a^-_{th} \ > = \ 1 - \ < a^+_{th} \ > \tag{21}$$

and

$$< A_{th} > = \ \frac{1}{2\,D_{tr}\,(h)} \ \left(e^{-\beta M^+_{th}} + e^{-\beta M^-_{th}} \right) \ , \tag{22}$$

where $M^{\pm}_{th}$ are the miss distances of the track from the two possible hit locations and $a^{\pm}_{th}$ are their respective assignment probabilities, with obvious changes in Eq.**??**.

IV. CLASS ORGANIZATION

Now we turn to a general overview of the classes, i.e., the type of objects, that will be introduced to carry out this fitting procedure and useful classifications that will be needed. One specific example of this classes is given in the next section.

Hits: This class is to contain the experimental information to be used in the analysis. The Hit class may also have to contain certain parameters used in the fitting procedure that characterize the particular hit only. These should be added by subclassing, but for expediency I have chosen not to do so below.

Group: This is the abstract class that is used as a foundation for the mathematical, or rather physical, constructs such as Track, DeltaRay, Vee, Kink, and others (such as jets, showers, etc.) that the user may add. This foundation class provides an outline that the user must use when adding a new class so that it is guaranteed to operate within the scheme.

Subclasses of Group: Each member of this set of classes, such as Track, Helix, Kink, etc., contain the proper number of parameters to define itself and also contains the auxiliary quantities and methods for fitting itself to the hits. These classes can contain data structures which are themselves members of the class; for example a Kink class object contains pointers to Track objects and its methods will utilize the fact that each Track object already has its version of the method defined. Among the parameters of this set of classes is an array of particle identification probabilities.

ParticleID and ParticleChain: The particle identification list contains standard particle types, such as γ , e , μ , π , ... The particleChain is used in the Kink and Vee classes to label a pair of associated Tracks, such as $\pi \,\&\, \pi$, $\pi \,\&\, \mu$, $\pi \,\&\, e$, $\mu \,\&\, e$, etc., in order to fit decay in flight, multiple scattering, neutral decays into a charged pair, ... In order to proceed with the particleID step, one must of course provide detector response information such as what a particular measured discriminate is expected to be for a particle of a listed type. This of course suggests introducing a detector response class to provide such information upon request.

V. IMPLEMENTATION IN OOP

In this section, an implementation of the elastic arms approach to track reconstruction will be described using OOP; C++ will be used for definiteness, but the important point is the use of classes and polymorphism to simplify the task of program design and implementation. The Hit class is defined as:

```
class Hit {
 public:
        Hit( );
       ~Hit( );
/* Public utility print & set/get methods go here.*/
  private:
   int  label;
   float z, h, dh, pulseHeight;
   float hcos,resolution;
   float (*hSag) (..);
   float Heff, Dtr;
   float noise, N;
   BOOL  owned;
   int   owner;
  }
```

where the meaning of label and z are clear.

The height of the hit is h, dh is the ambiguity distance and pulseHeight is itself. One could also include pulseShape variables if needed. The instance variable hcos is the cosine of the wire angle in the detector planes, such that h=x for hcos=1, and h=y for hcos=0; resolution is also obvious. The function pointer *hSag(..) corrects h for any sag in the wire and its arguments are the transverse coordinates of the track. Each wire plane will have its own sag function so this pointer is set at initialization of the data. All the following variables change during the fitting process; Heff is the value of the quantity $H_{\text{eff}}\ [h,\ \mathbf{T}\,]$ given in [**?**], and Dtr is the value of $D_{tr}\,(h)$. The owned and owner variables allow one to track the assignment of this hit to a group object. The abstract Group class is defined by the header file:

```
class Group {
 public:
           Group( );
  virtual ~Group( );
   virtual  void  setLabelGroup(int ,enum groupType );
   virtual  void  initialize(void) { };
   virtual  void    calcMiss(void) = 0;
   virtual  void  calcAssign(void) = 0;
   virtual  void  varyParams(void) = 0;
/* Many public utility print & set/get methods go here.*/
 protected:
   int    label;
   enum   groupType grp;
   static Hit **hit;
   static int numHits;
   static float beta, lambda, eta;
   float  zmin, zmax;
   float  missp[MAXHITS], missm[MAXHITS];
   float  AH[MAXHITS], ap[MAXHITS], am[MAXHITS];
   float  miss[MAXPARTICLETYPES];
   float  AP[MAXPARTICLETYPES];
   float  Heff, Did;
   float  ufo, N;
  }
```

where label is for bookkeeping convenience, and the enum variable groupType is { group, track, deltaRay, vee, kink, etc., } . By way of review, the static variables are used by all the members of the class Group (and its subclasses) during the fitting process. All members of the Group class physically extend over a finite extent from zmin to zmax; hits outside this range are given a miss distance of infinity. Each group object needs to know its miss distance from every hit position [5]; missp is the miss from (h+dh), and missm from (h-dh), the two possible hit positions. The array AH[h] is the assignment probability of hit h to this object, and the arrays ap[ ] and am[ ] are the relative probabilities for the hit positions (h ± dh). The new instance variables are miss[p], the miss distance from the particle type p, and the assignment variable AP . The quantity AP[p] is the probability that this group object is of the pth listed particle type. Heff is the value of the quantity $H_{\text{eff}}\ [t,\ \mathbf{T}\,]$ for this track t and Did is the value of $D_{id}\,(t)$. The probability

that this track cannot be classified is ufo, and finally N is the number of hits that this particular group object fits (owns).

The realizable subclasses of Group, such as Track, DeltaRay, Kink, etc., are defined and discussed in ref [1], as are the necessary Controller and Classifier classes. They need no further discussion here.

VI. CONCLUSIONS

The deformable template method has a very interesting mathematical structure with a clear physical interpretation. One of its most useful concepts are the neuron-type variables, denoted here by A_{th} and A_{pt} , that "assign" a hit to a track and a track to a particular particle type, respectively. They can be interpreted respectively as the probability (as assigned by the procedure and its inputs) that hit h belongs to track t and that the track t belongs to particle type p.

Note that after the detector properties are set, only the track class has parameters that can be varied. Thus one might question whether or not the second term in Eq.**??** has an effect on the final results. Consider a situation in which an alignment of hits is well fit geometrically as a straight track. It could also be equally well fit as a (straight) kink; if, however, the particle in question decayed forward in flight with no additional tracks, the kink class can provide a better fit by assigning a particle chain to the kink instance.

The important point here is that the template need not be purely geometric; it can also be extended to include particle type. One other useful feature of this algorithm is that it lends itself to encoding using object oriented programming techniques in a very natural way. The use of a object oriented programming language simplifies programming and its physical interpretation as discussed in [1]. Finally, the algorithm can be extended in obvious ways to a variety of pattern recognition and identification problems.

ACKNOWLEDGEMENTS

Many valuable conversations with Carsten Peterson and Mattias Ohlsson of Lund University are gratefully acknowledged as well as very helpful discussions with George Irwin of SLAC.

---

[1] *Deformable Templates—Revisited and Extended—with an OOP Implementation*, SLAC–PUB–6190, May 1993, submitted to *Computer Physics Communications*.

[2] M. Ohlsson, C. Peterson, A. Yuille, "Track Finding with Deformable Templates—The Elastic Arms Approach," *Computer Physics Communications* **71**, 77 (1992).

[3] M. Ohlsson, "Extensions and Explorations of the Elastic Arms Algorithm," Lund University Preprint LU-TP-92-28. To be published in *Computer Physics Communications*.

[4] For example, M. Gyulassy and H. Harlander, "Elastic Tracking and Neural Network Algorithms for Complex Pattern Recognition," *Computer Physics Communications* **66**, 31 (1991).

[5] It is not necessary to compute all of these distances. If one coordinate difference exceeds a set value, then the miss distance is assigned the value infinity.