;20140731

;Given a time-range and optional AR location, or tracking number

;Get Data

;Detect all regions

;Run tracking

;Pull out properties

pro run\_detect\_track\_prop, intstart, intend, cadence=incadence, aiawave=inaiawave, $

artracknum=artracknum, arpxpos=arpxpos, aridinit=aridint, $

nodetect=nodetect, indetsav=indetsav, outdetsav=outdetsav, $

notrack=notrack, intracksav=intracksav, outtracksav=outtracksav, $

noprop=noprop, inpropsav=inpropsav, outpropsav=outpropsav, $

outplot=outplot

paths=paths

tstart=anytim(intstart,/vms)

tend=anytim(intend,/vms)

if data\_type(paths) eq 8 then begin

root=paths.root

pdata=paths.pdata

fparam=paths.fparam

endif else begin

root='./'

pdata=root+'data/'

fparam='ar\_param\_ardetectmasks\_hmi.txt'

endelse

params=ar\_loadparam(fparam=root+fparam)

;Initialize housekeeping structure

strcsvdet={datafile:'',maskfile:'',date:'',tim:0l,nar:0,status:0}

if n\_elements(indetsav) eq 0 then detsav=pdata+'detected\_ars\_'+time2file(tstart)+'\_'+time2file(tend)+'.sav' $

else detsav=indetsav

if n\_elements(intracksav) eq 0 then tracksav=pdata+'tracked\_ars\_'+time2file(tstart)+'\_'+time2file(tend)+'.sav' $

else tracksav=intracksav

if n\_elements(inpropsav) eq 0 then propsav=pdata+'properties\_ars\_'+time2file(tstart)+'\_'+time2file(tend)+'.sav' $

else propsav=inpropsav

if n\_elements(incadence) ne 1 then cadence='1200s' else cadence=incadence

hmids='hmi.M\_720s'

aiads='aia.lev1\_euv\_12s'

if not keyword\_set(inaiawave) then aiawave=211 else aiawave=inaiawave

if not keyword\_set(nodetect) then begin

;List HMI Data---------------------------------------------------------------->

ssw\_jsoc\_time2data,tstart,tend,indhmi,fhmi,cadence=cadence,ds=hmids,/jsoc2,/files

;TEMP HACK!!!!!!!!!!!!!!!!!!!!!

;Check to make sure there are no file repeats

fdateind=indhmi.date\_obs

sdate=sort(fdateind)

fdateind=fdateind[sdate]

indhmi=indhmi[sdate]

fhmi=fhmi[sdate]

udate=uniq(fdateind)

fdateind=fdateind[udate]

indhmi=indhmi[udate]

fhmi=fhmi[udate]

;!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

;Detect ARs------------------------------------------------------------------->

nfile=n\_elements(fhmi)

for i=0,nfile-1 do begin

;Process and then Down-sample the 4k magnetogram to a 1k

thismap4k=ar\_readmag(fhmi[i],indhmi[i], outindex=ind)

if i eq 0 then hmiindarr=indhmi[i] else hmiindarr=[hmiindarr,indhmi[i]]

thismap4kp=ar\_processmag(thismap4k,param=params, maxlimb=params.maxlimbdetect,/nocosmicray,/nofilter)

thismap=map\_rebin(thismap4kp,/rebin1k)

;Create AR Core mask (includes processing of MDI image -> read out into THISPROC)

thissmstr=ar\_detect(thismap, params=params, status=smartstatus, cosmap=cosmap, limbmask=limbmask) ;,/nocosmic)

thisarstr=ar\_detect\_core(thismap, smartmask=thissmstr.data, maxlimb=params.maxlimbdetect, params=params, status=corestatus, cosmap=cosmap, limbmask=limbmask, pslmaskmap=pslmap); ,/nocosmic)

thismask=ar\_core2mask(thisarstr.data)

;Make array of maps and magnetograms for later tracking

if i eq 0 then maskmaparr=thisarstr else maskmaparr=[maskmaparr,thisarstr]

if i eq 0 then magmaparr=thismap else magmaparr=[magmaparr,thismap]

if i eq 0 then pslmaparr=pslmap else pslmaparr=[pslmaparr,pslmap]

;Make all maps 4kx4k again

imgszorig=size(thismap4k.data,/dim)

thisarstr4k=thismap4k

thisarstr4k.data=round(congrid(thisarstr.data,imgszorig[0],imgszorig[1]))

pslmap4k=thisarstr4k

pslmap4k.data=round(congrid(pslmap.data,imgszorig[0],imgszorig[1]))

pslmask4k=pslmap4k.data

;Make indexed core mask

coremask4k=ar\_core2mask(thisarstr4k.data,smartmask=coresmblob4k,coresmartmask=coresmblob\_conn4k)

thisdatafile=time2file(indhmi[i].date\_obs)

help,thisdatafile

pospropstr=ar\_posprop(map=thismap, mask=thismask, cosmap=cosmap, params=params, $

outpos=outpos, outneg=outneg, /nosigned, status=posstatus, datafile=thisdatafile)

npos=n\_elements(pospropstr)

thisstrcsvdet=strcsvdet

thisstrcsvdet.datafile=thisdatafile

thisstrcsvdet.maskfile='smart\_core\_'+time2file(indhmi[i].date\_obs)+'.fits'

thisstrcsvdet.date=anytim(indhmi[i].date\_obs,/vms)

thisstrcsvdet.tim=anytim(thisstrcsvdet.date)

thisstrcsvdet.nar=max(coremask4k)

thisstrcsvdet.status=corestatus

if i eq 0 then posproparr=pospropstr else posproparr=[posproparr,pospropstr]

if n\_elements(detstrarr) eq 0 then detstrarr=thisstrcsvdet else detstrarr=[detstrarr,thisstrcsvdet]

plot\_mag,thismap4kp.data

contour,coremask4k,lev=0.5,/over

endfor

;Make detection structure correspond to

arstrarr=ar\_detstr2arstr(detstrarr,posproparr.datafile)

combine\_structs,arstrarr,posproparr,smartmeta

;Save detection data---------------------------------------------------------->

save,smartmeta,posproparr,detstrarr,maskmaparr,magmaparr,pslmaparr,fhmi,tstart,tend, indhmi, file=detsav,/comp

outdetsav=detsav

endif else begin

restore,detsav

endelse

;Make maps of the masks

maskdatarr=maskmaparr.data

maskdatarr=ar\_core2mask(maskdatarr)

maskmaparr.data=maskdatarr

if not keyword\_set(notrack) then begin

;Track ARs-------------------------------------------------------------------->

mdimeta=detstrarr

undefine,state

trackstr=ar\_track\_yafta(state,magstack=magmaparr.data,maskstack=maskmaparr.data, $

mdimeta=mdimeta,smartmeta=smartmeta, $

params=params, /doplot, $

outsingle=outsingle)

;Choose ARs------------------------------------------------------------------->

!p.multi=0

plot\_mag,magmaparr[0].data

contour,maskmaparr[0].data,level=0.5,/over

wshow

print,outplot

if n\_elements(artracknum) eq 0 then begin

if n\_elements(arpxpos) eq 0 then begin

print,'CLICK TO CHOOSE AR CONTOUR TO TRACK!!!'

cursor,arpxx,arpxy,/data

endif else begin

arpxx=arpxpos[0]

arpxy=arpxpos[1]

endelse

aridint=(maskmaparr[0].data)[arpxx,arpxy]

endif

;Save tracking data---------------------------------------------------------->

save,aridint,trackstr,file=tracksav,/comp

outtracksav=tracksav

endif else restore,tracksav

;Pick out elements corresponding to AR of interest

wthisarid=where(trackstr.arid eq aridint)

thisyaftaid=trackstr[wthisarid[0]].YAFTAID

wthisyaftaid=where(trackstr.YAFTAID eq thisyaftaid)

posproparr=posproparr[wthisyaftaid]

;Plot first and last contour in series, with tracked pos. overlayed

plot\_mag, magmaparr[0]

plot\_map, maskmaparr[0], level=0.5,/over

plot\_map, maskmaparr[n\_elements(maskmaparr)-1], level=0.5,/over

;plot,posproparr.HGLONFLX,posproparr.HGlatFLX,ps=-4,xran=[-90,90],yran=[-90,90]

oplot,posproparr.hcxFLX,posproparr.hcyFLX,ps=4;,xran=[-1100,1100],yran=[-1100,1100]

;Characterise ARs------------------------------------------------------------->

cutbuffer=20

ntrack=n\_elements(wthisyaftaid)

if not keyword\_set(noprop) then begin

for i=0,ntrack-1 do begin

thismap4k=ar\_readmag(fhmi[i],indhmi[i], outindex=ind)

thismap4k=rot\_map(thismap4k,-thismap4k.roll\_angle)

imgsz4k=size(thismap4k.data,/dim)

thismap=magmaparr[i]

thismask=maskmaparr[i].data

;Crop around the AR in question

thistrackstrar=trackstr[wthisyaftaid[i]]

thisid=thistrackstrar.arid

thismask[where(thismask ne thisid)]=0

thismask[where(thismask eq thisid)]=1

;!!!PLOT THIS MASK TO DISPLAY PROGRESS

plotthismask=maskmaparr[i]

plotthismask.data=thismask

plot\_map,plotthismask,level=0.5,/over

;!!!

;expand the mask to fit the 4k res image

thismask4k=round(congrid(smooth(thismask,[4,4]),imgsz4k[0],imgsz4k[1],/inter))

; thismask4k=round(congrid(thismask,imgsz4k[0],imgsz4k[1]))

maskmap4k=thismap4k & maskmap4k.data=thismask4k

;Make sub-maps

sub\_map,maskmap4k,submask,xran=minmax(where(maskmap4k.data eq 1) mod imgsz4k[0])+[-cutbuffer,cutbuffer],yran=minmax(where(maskmap4k.data eq 1)/imgsz4k[0])+[-cutbuffer,cutbuffer],/pixel,/noplot

sub\_map,thismap4k,submag,ref=submask,/noplot

map2wcs,submask,wcsmask & add\_prop,submask,wcs=wcsmask,/repl

map2wcs,submag,wcsmag & add\_prop,submag,wcs=wcsmag,/repl

;Process the magnetogram

submag=ar\_processmag(submag,/nofilt,/nocosmic)

;Find AIA file

ssw\_jsoc\_time2data,thismap.time,anytim(anytim(thismap.time)+3600.),indaia,faia,cadence='3600s',ds=aiads,wave=aiawave,/jsoc2,/files

read\_sdo,faia[0],inde,date,/nosh,/useshare

aia\_prep, indaia[0], date, indep, datep

mindex2map,indep,datep,mape

;!!!!!!WRITE A AR\_READAIA() function to rotate, get rid of nans, float everything, divide out exposure control, etc

;Convert AIA data to floats

add\_prop,mape,data=float(mape.data),/repl

mape.data=mape.data/mape.exptime

;Crop each mask down to the AR

sub\_map,mape,submape,ref=submask,/noplot

map2wcs,submape,wcsmape & add\_prop,submape,wcs=wcsmape,/repl

; rbsubmapedat=congrid(submape.data,szsubmp[0],szsubmp[1])

; submaperb=submag

; submaperb.data=rbsubmapedat

;Then determine properties for tracked AR over time

magpropstr=ar\_magprop(map=submag, mask=submask, cosmap=subcosmap, params=params, status=magstatus, datafile=smartmeta[wthisyaftaid[i]].datafile)

;Make array of property structures

if i eq 0 then magproparr=magpropstr else magproparr=[magproparr,magpropstr]

;match MAG and AIA img sizes (shrink MAG)

szsubmp=size(submape.data,/dim)

submagaia=submape

submagaia.data=congrid(submag.data,szsubmp[0],szsubmp[1],/interp)

;Pull out PSL skeleton to measure shear across it

pslmask=ar\_pslmask(ar\_grow(submagaia.data,rad=5,/gaus),radius=5\*2.,thresh=200,/dothin)

pslmask=congrid(pslmask,szsubmp[0],szsubmp[1])

subpslt=submape

subpslt.data=pslmask

;CODE TO PULL OUT AIA PROPERTIES

thismaskaia=round(congrid(smooth(submask.data,[4,4]),szsubmp[0],szsubmp[1],/inter))

euvstr = ar\_euvprop(map=submape, mask=thismaskaia, pslmask=subpslt.data, params=params, datafile=thistrackstrar.datafile, outloopangmap=loopangmap, outplotstr=outplotstr)

if n\_elements(euvstrarr) eq 0 then euvstrarr=euvstr else euvstrarr=[euvstrarr,euvstr]

;Make a movie of the region

if n\_elements(outplot) eq 1 then begin

set\_plot, 'z'

resxy=[2100,700]

thisimg=outplot+'\_'+strtrim(string(i,form='(I04)'),2)+'.png'

device, set\_resolution = resxy, decomp=0

device, set\_pixel\_depth=24

erase

!p.multi=[3,3,1]

loadct,0

plot\_mag,submag,/nobg

setcolors,/sys,/sil

plot\_map,subpslt,level=0.5,c\_color=!red,/over

aia\_lct,wave=211

!p.multi=[2,3,1]

plot\_map,submape,/log,dran=[100,10000],/noerase

setcolors,/sys,/sil

!p.multi=[2,3,1]

plot\_map,subpslt,level=0.5,c\_color=!white,/over

; loadct,0,/sil

; !p.multi=[1,3,1]

; loadct,39,/sil

; plot\_image,abs(loopangmap/2.)/!dtor,/noerase

; angmm=minmax(abs(loopangmap/2.)/!dtor)

; color\_table, angmm, [0.7,0.99],[0.1,0.15],color=255,chars=3,charth=1,xtit='Loop Angle [deg]'

!p.multi=[1,3,1]

plot\_image,outplotstr.image,true=3,/noerase

;plot,[0,outplotstr.nax(0)],[0,outplotstr.nax(1)],/nodata,/noerase;,xsty=5,ysty=5,xmar=[0,0],ymar=[0,0]

;for i=0l,outplotstr.ngrid-1l do plots,(outplotstr.xarrow)[i],((outplotstr.yarrow)[i])\*[-1,1],col=0,thick=5

;for i=0l,outplotstr.ngrid-1l do plots,(outplotstr.xarrow)[i],((outplotstr.yarrow)[i])\*[-1,1]

xyouts,0.05,0.05,time2file(submag.time),/norm,chars=3

xyouts,0.05,0.95,outplot,/norm,chars=3

zb\_plot=tvrd(true=1)

write\_png, thisimg, zb\_plot

set\_plot, 'x'

endif

; pslpropstr=ar\_pslprop(thismap, thisarmap, param=params, $

; /doproj, projscl=projscl, /dobppxscl, $

; outproj=thisprojmag, outscl=thisprojpxscl, outbpscl=thisprojpxscl\_bpsep, outmaskproj=thisprojmask, $

; outrefproj=thisprojref, projlimbxy=projlimbxy, outpslmask=pslmaskt, outgradpsl=gradpsl, projmaxscale=projmaxscale)

endfor

save,euvstrarr,magproparr,posproparr,detstrarr,maskmaparr,magmaparr,pslmaparr,fhmi,tstart,tend,file=propsav,/comp

outpropsav=propsav

endif else restore,propsav,/ver

;Make plot

if n\_elements(outplot) eq 1 then begin

set\_plot, 'z'

resxy=[1200,1400]

thisimg=outplot+'\_timeseries.png'

device, set\_resolution = resxy, decomp=0, set\_pixel\_depth=24

endif else window,xs=1200,ys=1400

setcolors,/sys,/sil

mintim=min(detstrarr.tim)

!x.margin=[15,15]

!y.margin=[2,2]

!p.multi=[0,1,7]

utplot,detstrarr.tim-mintim,magproparr.totflx,mintim,ps=-4,chars=2,/ysty,ytit='TOTAL FLUX',/xsty

vline,anytim('1-jul-2012 15:43')-mintim,lines=2,xtit=''

utplot,detstrarr.tim-mintim,magproparr.frcflx,mintim,ps=-4,chars=2,/ysty,ytit='FRAC IMB',xtit='',/xsty

utplot,detstrarr.tim-mintim,euvstrarr.MAXINTENSE,mintim,ps=-4,chars=2,/ysty,ytit='MAX INT',xtit='',/xsty

!p.multi=[4,1,7]

utplot,detstrarr.tim-mintim,euvstrarr.TOTALINTENSE,mintim,ps=-4,chars=2,/ysty,ytit='TOT INT',xtit='',/noerase,/xsty

utplot,detstrarr.tim-mintim,euvstrarr.MEANINTENSE,mintim,ps=-4,chars=2,yran=minmax(euvstrarr.MEANINTENSE),yticklen=0.0001,ytickname=strarr(10)+' ',color=150,/noerase,/xsty ;,ytit='MEAN INT',xtit=''

axis,/yaxis,yran=minmax(euvstrarr.MEANINTENSE),color=150,chars=2,ytit='MEAN INT'

!p.multi=[3,1,7]

utplot,detstrarr.tim-mintim,euvstrarr.LOOPtotPSL,mintim,chars=2,yran=minmax(euvstrarr.LOOPtotPSL),/ysty,color=150,ps=-4,yticklen=0.0001,ytickname=strarr(10)+' ',xtit='',/noerase,/xsty

utplot,detstrarr.tim-mintim,euvstrarr.LOOPMEDIANSHEAR,mintim,ps=-4,chars=2,/ysty,ytit='<LOOP/PSL ANG>\_wgt',xtit='',/noerase,/xsty

axis,/yaxis,yran=minmax(euvstrarr.LOOPtotPSL),color=150,chars=2,ytit='Tot. Len. PSL'

!p.multi=[2,1,7]

utplot,detstrarr.tim-mintim,euvstrarr.LOOPtotPSLlrg,mintim,chars=2,yran=minmax(euvstrarr.LOOPtotPSLlrg),/ysty,color=150,ps=-4,yticklen=0.0001,ytickname=strarr(10)+' ',xtit='',/noerase,/xsty

utplot,detstrarr.tim-mintim,euvstrarr.LOOPMEDIANSHEARlrg,mintim,ps=-4,chars=2,/ysty,ytit='Lrg. <LOOP/PSL ANG>\_wgt',xtit='',/noerase,/xsty

axis,/yaxis,yran=minmax(euvstrarr.LOOPtotPSLlrg),color=150,chars=2,ytit='Lrg. Tot. Len. PSL'

;Plot the flares light curve

!p.multi=[1,1,7]

goesobj=ogoes()

goesobj->set,tstart=tstart,tend=tend

goesobj->plot,/noerase,xmargin=!x.margin,ymargin=!y.margin,chars=2,/xsty ;,position=[0.1,0.05,0.95,0.3],xstyle=1

; evt\_grid,dates[i],thick=5,lines=2

if n\_elements(outplot) eq 1 then begin

zb\_plot=tvrd(true=1)

write\_png, thisimg, zb\_plot

set\_plot, 'x'

endif

stop

return

end