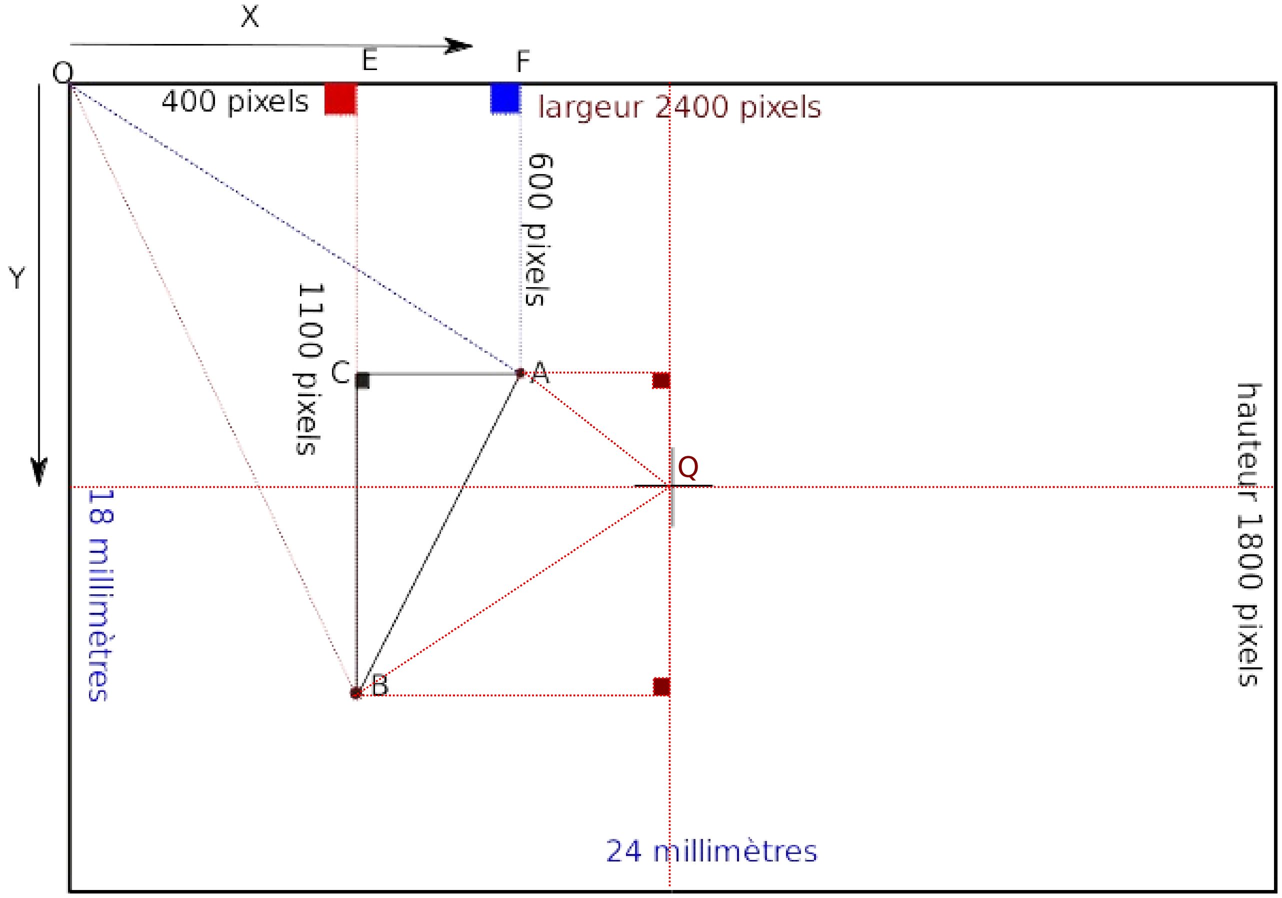
DE QUEL ANGLE L'OBJET S'EST-IL DÉPLACÉ ?

Voici une question qui m'a été posée il y a peu et je voudrais vous faire profiter du raisonnement suivant.
Si vous le trouvez discutable, tant mieux : discutons-en ?

1. Le document représenté

Il s'agit d'une simplification d'un document résultant de l'addition de deux photos d'un astéroïde en déplacement rapide. Ce pourrait-être une comète…
On supposera par la suite que l'alignement des photos a été parfait et que le document obtenu restitue la réalité, et que l'objectif utilisé n'introduit aucune distorsion.
L'image de l'astéroïde a occupé successivement les points A et B.

Le capteur de l'appareil dispose d'une définition de 2400 pixels sur 1800 pixels.
Sa dimension physique est de 24 millimètres sur 18 millimètres.
Le point marqué O est l'origine des axes (y et x) pour les images JPEG.
Le point A est à 600 pixels du bord supérieur de l'image (la distance marquée FA) et à 700 pixels du bord gauche de l'image (distance O F). La mesure n'est pas portée sur le dessin.
Le point B est à 1100 pixels du bord supérieur de l'image (la distance marquée BE) et à 400 pixels du bord gauche de l'image.
Les angles marqués d'un carré sont des angles droits.
A partir de la position en A et en B, j'ai reconstitué le triangle rectangle ABC.
Le côté AC mesure 700 – 400 = 300 pixels. Le côté BC mesure 1100 – 600 = 500 pixels.



2. Equivalence pixel – millimètre

Si 2400 pixels représentent 24 millimètres, un pixel représente 24 / 2400 = 1/100 ème de millimètre ou 0,01 millimètre. On peut faire le même calcul sur la hauteur et l'on trouvera la même valeur.

Segment	Distance en pixels	Distance en millimètres
AC	300	3
BC	500	5

Mais quelle est la mesure de AB en millimètres ?
Si vous vous souvenez de la relation de Pythagore, vous retrouverez ceci :
AB au carré = AC au carré + BC au carré → AB x AB = 3x3 + 5 x 5 → AB = racine carrée de 34.
Une calculette donne AB = 5, 83… millimètres.

3. Mais comment en déduire un déplacement angulaire ?

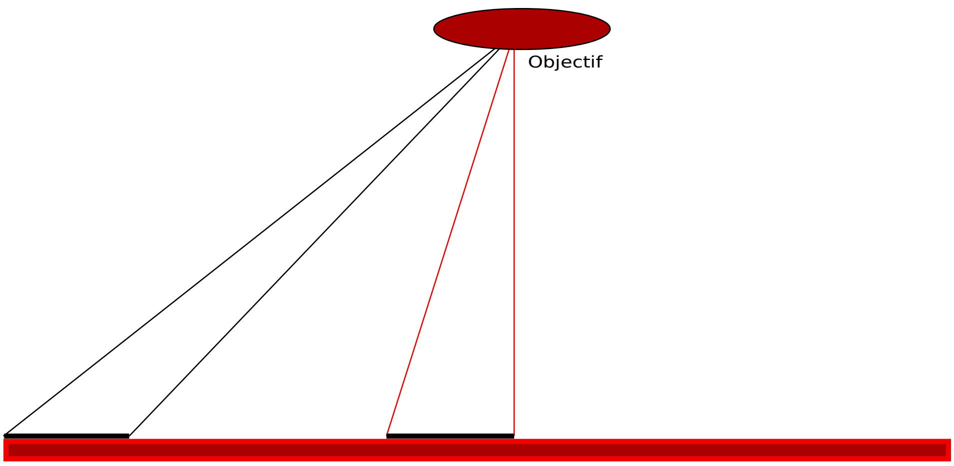
Il nous manque une information capitale : la longueur focale de l'objectif.
Celle-ci est de 100 millimètres.
Sous réserve de se situer très près du centre du cliché (la croix sur le dessin), l'angle sous-tendu par un écart de 1 millimètre sur le capteur (un triangle rectangle ayant pour côtés 100 et 1) est donné par :
ARC TANGENTE de 1/100
Le moteur de recherche de Google donne arctangente(0,01) = 0,00999966669 rad
Et l'angle en radian (la signification de « rad » dans la réponse de Google) peut être converti en degrés en multipliant 180 puis en divisant le résultat par Pi (3,1416…)
0,00999 * 180 / 3,1416 = 0,572383499 degré (soit un peu plus de 1/2 degré).

Ceux qui ont suivi se frottent les mains : si j'applique le même raisonnement à 5,83 millimètre qu'à 1 millimètres, j'aurais ma réponse.
Essayons : 5,83 / 100 = 0,0583
arctangente(0,0583) donne 0,05624 radians
0,05624 * 180 / 3,1416 = 3,222… degrés

Chouette : j'ai ma réponse : 3,222… degrés

4. Mais il y a rabat-joie...

Hélas, ce n'est pas si simple. Souvenez-vous de la remarque en rouge.
Dans l'idéal, il faudrait que le segment AB passe par le centre du cliché et notre réponse serait plus proche de la réalité.



Le dessin ci-dessus illustre la cause de l'incertitude : une même longueur (segment noir) sur le capteur rouge, couvre un angle plus large vers le centre (tracé rouge) que vers le bord (tracé noir).
Ce n'est pas pour rien que les calculs astronomiques ont la réputation d'être un peu compliqués.

Disons que cette note aura donné quelques éléments pour poser le problème.
Et pour la distance angulaire AB je peux écrire que c'est autour de 3 degrés mais moins que 3,22 degrés.

5. Avant de discuter d'une solution plus exacte un peu de code Python

(Les personnes que le codage n'intéressent pas peuvent passer au point 6).
L'occasion est trop belle de donner un exemple de programme Python sur ce cas simple.

D'abord le code source :

```
# -*- coding:Utf-8 -*-
from math import sqrt, atan, pi

"""
    Distance angulaire entre deux points d'une photos numérique
    Alain LERAUT - février 2016
    Sur une idée de Gérad Cloarec
"""
"""          valeurs liées à l'appareil photographique             """
larg_en_pixels   = 2400
haut_en_pixels   = 1800
larg_en_mm       = 24.0 # en valeur décimale
haut_en_mm       = 18.0 # en valeur décimale
focale              = 100
"""  position des centres des deux objet en pixels
    cette position est déterminée par rapport au coin haut-gauche """
xA_pixels        = 700
yA_pixels        = 600
xB_pixels        = 400
yB_pixels        = 1100

"""  zone des calculs simplifiés et très approximatifs"""

eq_pix_mm    =   larg_en_mm / larg_en_pixels
print("Equivalent pixel-millimètre : {} ".format(eq_pix_mm))
AC = (xA_pixels - xB_pixels) * eq_pix_mm
BC = (yA_pixels - yB_pixels) * eq_pix_mm
AB = sqrt(AC**2 + BC**2)

print("Valeur de AC : {} millimètres".format(AB))
rapport = AB / focale
angle_rad = atan(rapport)
print("Angle à peu près : {} radians".format(angle_rad))
print("Angle à peu près : {} degrés".format(angle_rad* 180 / pi))
```

Quelques commentaires.

`# -*- coding:Utf-8 -*-` pour avoir accès aux caractères français

`from math import sqrt, atan, pi` on importe les fonctions mathématiques utiles
(remarques : de façon à réduire la taille du programme on n'importe que ce dont on a besoin)

`"""` tout ce qui est ici est un commentaire `"""`

`larg_en_pixels    = 2400` Les valeurs sont associées à un nom, ce qui permet leur manipulation

`print("Angle à peu près : {} degrés".format(angle_rad* 180 / pi))` manière d'afficher

Ce qu'affiche le programme à la fin de son exécution :

```
Equivalent pixel-millimètre : 0.01
Valeur de AB : 5.83095189485 millimètres
Angle à peu près : 0.058243569312 radians
Angle à peu près : 3.33711070536 degrés
```

6. Un formulaire pour un calcul plus juste

Formulaire utilisé :

En fait, il faut utiliser un formulaire permettant de connaître un angle d'un triangle quelconque dont on connaît la mesure des trois côtés.
Pour se raffraichir la mémoire (ou découvrir l'ampleur de son ignorance), on pourra lire ceci qui comprend des exemples d'application : http://www.warmaths.fr/MATH/geometr/TRIGONOMETRIE/reltrigotriqcq.htm#cas3

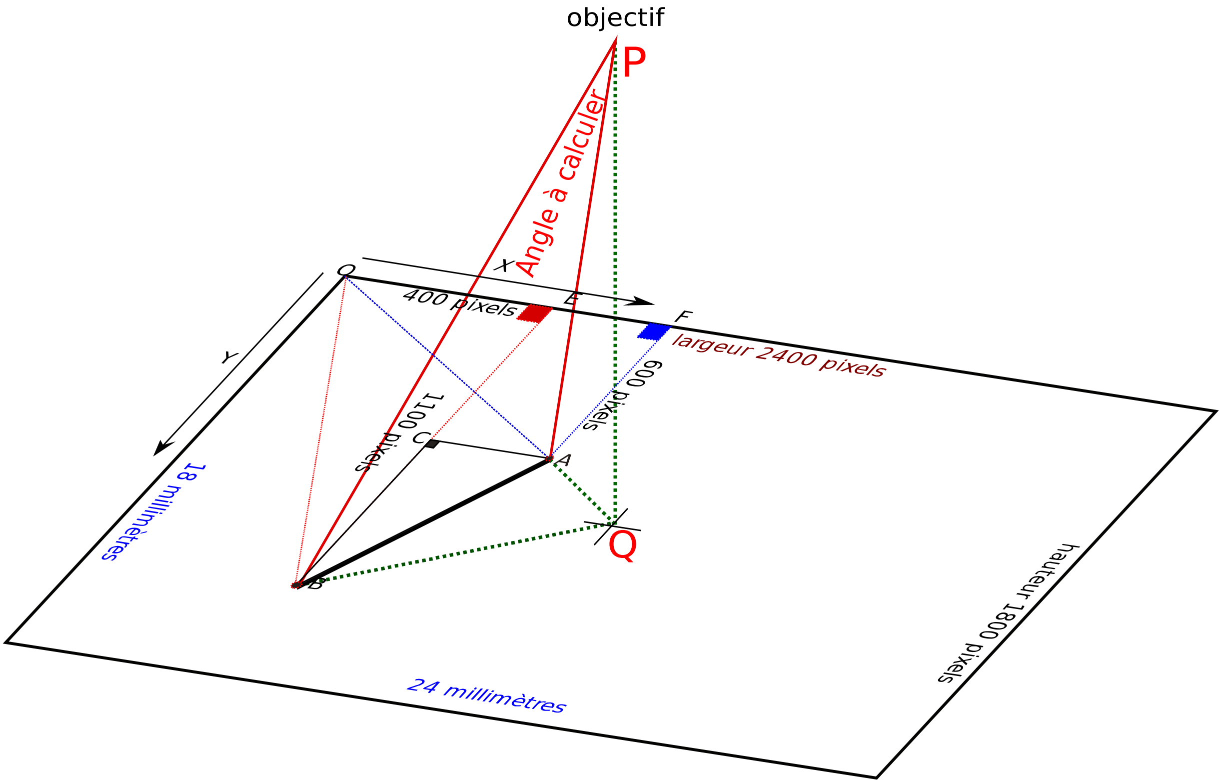
Vérification du formulaire via un programme en Python (les personnes réfractaires peuvent passer au point 7).
Avant de l'appliquer, il est utile de vérifier que tout se passe comme prévu.

```
a = 118.0
b = 65.90
c = 80.55
print("Les exemples du formulaire en ligne")
cos_A  = (b**2 + c**2 - a**2) / (2 * b * c)
print("Cosinus de A      :{}  ".format(cos_A))
A_rad = acos(abs(cos_A))
print("Angle A           : {} degrés ".format(A_rad *180 / pi))
sin_B = (b * sin(A_rad)) / a
print("Valeur de sinus B : {} ".format(sin_B))
B_rad = asin(abs(sin_B))
print("Angle B           : {} degrés ".format(B_rad * 180/pi))
```

Rien à en dire si ce n'est l'utilisation de la fonction abs pour rester dans le bon quadrant.

7. Une image pour comprendre la suite.

Le dessin est maintenant en perspective de façon à montrer les nouveaux éléments. Le point P est le centre de l'objectif et le point Q est le centre du capteur.



Pour déterminer l'angle qui nous intéresse, via le formulaire sur **les triangles quelconques**, il nous faut connaître les dimensions du triangle PAB. Nous avons AB = `5.83095189485 millimètres`

Le triangle PAQ est rectangle. Connaissant AQ et PQ on pourra déterminer AP.
Or AQ peut se déterminer par la formule de Pythagore.
Le triangle PBQ est rectangle. Connaissant BQ et PQ on pourra déterminer PB.
Or BQ peut se déterminer par la formule de Pythagore.

Nous avons besoin de deux valeurs nouvelles :
- dist_xpix_centre = la largeur en pixels divisée par 2 (en pixels). Ici, c'est 2400 / 2 = 1200
- dist_ypix_centre = la hauteur en pixels divisée par 2 (en pixels). Ici, c'est 1800 / 2 = 900

Calcul de AQ :
AQ = racine carrée de ((dist_xpix_centre – xA_pixels) au carré + (dist_ypix_centre – yA_pixels) au carré)
AQ = racine carrée de (1200 – 700) au carré + (900 – 600) au carré)
AQ = racine carrée de (500 x 500 + 300 x 300) = 583,09… pixels ou 5.83095189485 millimètres
Calcul de BQ :
BQ = racine carrée de ((dist_xpix_centre – xB_pixels) au carré + (dist_ypix_centre – yB_pixels) au carré)
AQ = racine carrée de (1200 – 400) au carré + (900 – 1100) au carré)
AQ = racine carrée de (800 x 800 + -200 x -200) = 583,09… pixels ou 8,24... millimètres

AP = racine carrée de (AQ au carré + PQ au carré) = 100.169855745 millimètres
BP = racine carrée de (BQ au carré + PQ au carré) = 100.339423957 millimètres

L'application du formulaire du triangle quelconque permet de calculer l'angle souhaité.
Le résultat est : **3.33146479887 degrés**
La suite du code du programme Python donne le formulaire appliqué.

```
dist_xpix_centre = larg_en_pixels / 2
dist_ypix_centre = 1800 / 2
AQ = sqrt(((dist_xpix_centre - xA_pixels)**2) + ((dist_ypix_centre - yA_pixels)**2))
AQ = AQ * eq_pix_mm
print("AQ                : {} en millimètres".format(AQ))
BQ = sqrt(((dist_xpix_centre - xB_pixels)**2) + ((dist_ypix_centre - yB_pixels)**2))
BQ = BQ * eq_pix_mm
print("BQ                : {}  en millimètres".format(BQ))
AP = sqrt(AQ ** 2 + focale ** 2)
BP = sqrt(BQ ** 2 + focale ** 2)
print("Distance du centre de l'objectif au point A (AP) : {} millimètres".format(AP))
print("Distance du centre de l'objectif au point B  'BP) : {} millimètres".format(BP))

""" --- on réaffecte les valeurs dans l'exemple du formulaire et on applique les
formules """

a = BP
b = AB
c = AP
print
print("Application du formulaire en ligne")

cos_A  = (b**2 + c**2 - a**2) / (2 * b * c)
print("Cosinus de A      :{}  ".format(cos_A))
A_rad = acos(abs(cos_A))
sin_B = (b * sin(A_rad)) / a
B_rad = asin(abs(sin_B))
print("Angle B           : {} degrés ".format(B_rad * 180/pi))
```

Et les résultats affichés sont :

```
Application du formulaire en ligne
Cosinus de A      :-1.55712567806e-15
Angle B           : 3.33146479887 degrés
```

En cas de désaccord, merci de me signaler les erreurs (ou de proposer d'autres solutions).